MC68HC705C4A/D

HC05

MC68HC705C4A

TECHNICAL
DATA

MOTOROLA

MC68HC705C4A
MC68HSC705C4A
HCMOS MICROCONTROLLER UNIT

MOTOROLA

CSIC

MICROCONTROLLERS

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

TABLE OF CONTENTS

Paragraph — Title — Page

SECTION 1 GENERAL DESCRIPTION

SECTION 2 MEMORY

SECTION 3 CENTRAL PROCESSOR UNIT (CPU)

TABLE OF CONTENTS

Paragraph	Title	Page

SECTION 4
INTERRUPTS

SECTION 5
RESETS

SECTION 6
LOW-POWER MODES

SECTION 7
PARALLEL INPUT/OUTPUT (I/O)

TABLE OF CONTENTS

Paragraph — Title — Page

SECTION 8
CAPTURE/COMPARE TIMER

SECTION 9
EPROM/OTPROM (PROM)

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 10
SERIAL COMMUNICATIONS INTERFACE (SCI)

SECTION 11
SERIAL PERIPHERAL INTERFACE (SPI)

SECTION 12
INSTRUCTION SET

TABLE OF CONTENTS

SECTION 13
ELECTRICAL SPECIFICATIONS

SECTION 14
MECHANICAL SPECIFICATIONS

SECTION 15
ORDERING INFORMATION

APPENDIX A
MC68HSC705C4A

INDEX

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

SECTION 1
GENERAL DESCRIPTION

1.1 Introduction

The MC68HC705C4A is similar to the MC68HC705C8A. The major differences are that the MC68HC705C4A has less RAM and ROM, no selectable memory configurations, no clock monitor reset function, and only one computer operating properly (COP) watchdog timer. The MC68HSC705C4A, introduced in Appendix A, is an enhanced, high-speed version of the MC68HC705C4A.

The MC68HC705C4A is a member of the low-cost, high-performance M68HC05 Family of 8-bit microcontroller units (MCUs). The M68HC05 Family is based on the customer-specified integrated circuit (CSIC) design strategy. All MCUs in the family use the M68HC05 central processor unit (CPU) and are available with a variety of subsystems, memory sizes and types, and package types.

1.2 Features

Features of the MC68HC705C4A include:

- M68HC05 CPU
- On-Chip Oscillator with Crystal/Ceramic Resonator
- Memory-Mapped Input/Output (I/O)
- Computer Operating Properly (COP) Watchdog Timer
- Selectable Port B External Interrupt Capability
- High Current Drive on Pin C7 (PC7)
- 24 Bidirectional I/O Lines and 7 Input-Only Lines
- Serial Communications Interface (SCI) System
- Serial Peripheral Interface (SPI) System
- Bootstrap Capability
- Power-Saving Stop, Wait, and Data-Retention Modes
- Single 3.0- to 5.5-Volt Supply (2-Volt Data-Retention Mode)
- Fully Static Operation
- Software-Programmable External Interrupt Sensitivity
- Bidirectional $\overline{\text{RESET}}$ Pin

NOTE

A line over a signal name indicates an active low signal. For example, RESET is active high and $\overline{\text{RESET}}$ is active low. Any reference to voltage, current, or frequency specified in the following sections will refer to the nominal values. The exact values and their tolerance or limits are specified in **SECTION 13 ELECTRICAL SPECIFICATIONS.**

1.3 Programmable Options

The following options are programmable in the mask option registers:

- Enabling of port B pullup devices (See **9.4.2 Mask Option Register 1 (MOR1)**.)
- Enabling of non-programmable COP watchdog (See **9.4.3 Mask Option Register 2 (MOR2)**.)

The following options are programmable in the option register (shown below in Figure 1-1):

- PROM security
- External interrupt sensitivity

		Bit 7	6	5	4	3	2	1	Bit 0
Option $1FDF	Read: Write:	0	0	0	0	SEC*		IRQ	0
	Reset:	0	0	0	0	*	U	1	0

▨ = Unimplemented   U = Unaffected

*Implemented as an EPROM cell

Figure 1-1. Option Register (Option)

SEC — Security

This bit is implemented as an EPROM cell and is not affected by reset.
1 = Bootloader disabled; MCU operates only in single-chip mode.
0 = Security off; bootloader able to be enabled.

IRQ — Interrupt Request Pin Sensitivity

IRQ is set only by reset, but can be cleared by software. This bit can be written only once.

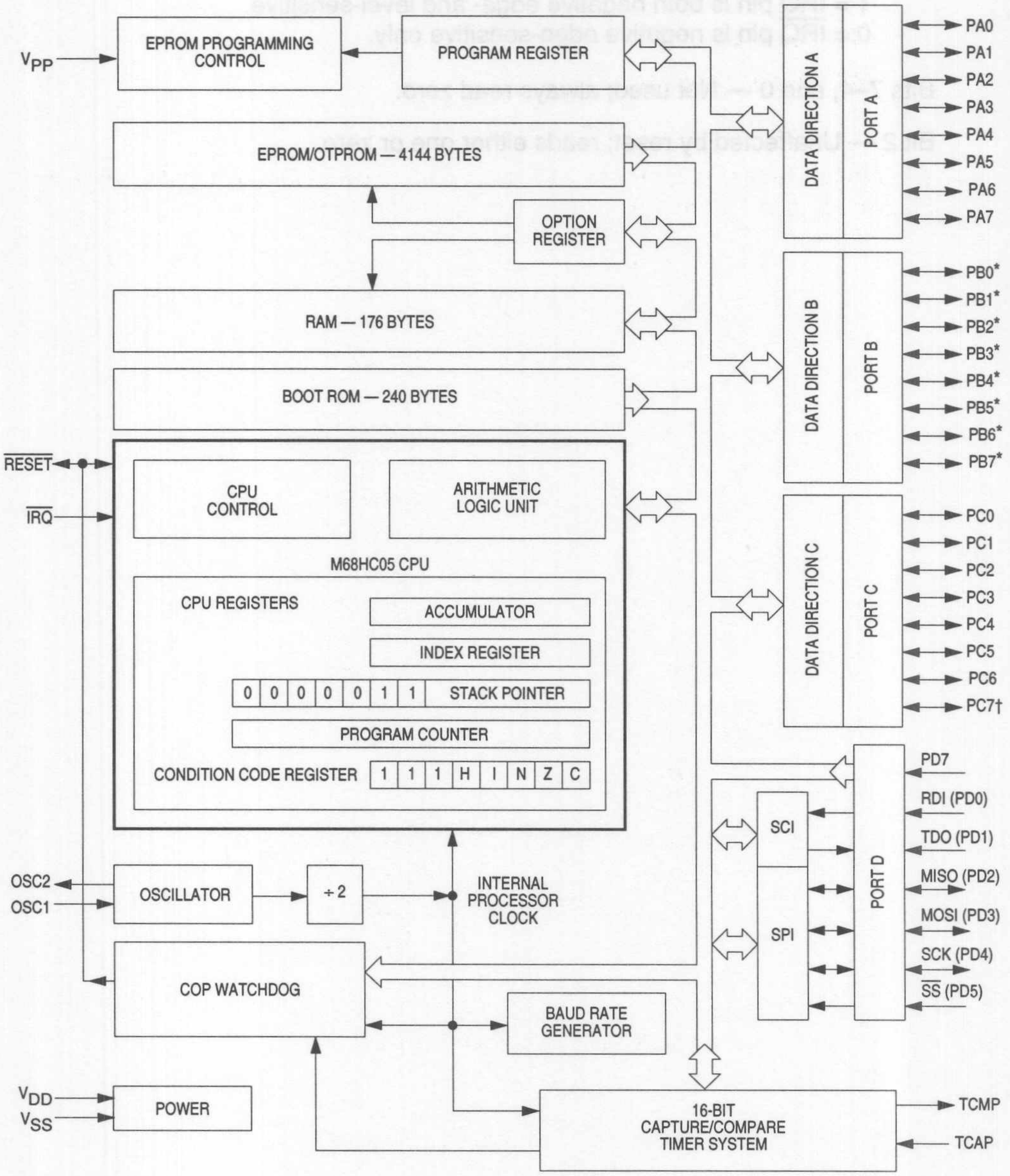
1 = $\overline{\text{IRQ}}$ pin is both negative edge- and level-sensitive.
0 = $\overline{\text{IRQ}}$ pin is negative edge-sensitive only.

Bits 7–4, and 0 — Not used; always read zero.

Bit 2 — Unaffected by reset; reads either one or zero.

1.4 Block Diagram

Figure 1-2 shows the structure of the MC68HC705C4A.



* Port B pins also function as external interrupts.
† PC7 has a high current sink and source capability.

Figure 1-2. Block Diagram

1.5 Pin Assignments

The MC68HC705C4A is available in the following packages:

- 40-pin plastic dual in-line package (PDIP)
- 44-lead plastic-leaded chip carrier (PLCC)
- 44-pin quad flat pack (QFP)
- 42-pin shrink dual in-line package (SDIP)

The pin assignments for these packages are shown in Figure 1-3, Figure 1-4, Figure 1-5, and Figure 1-6.

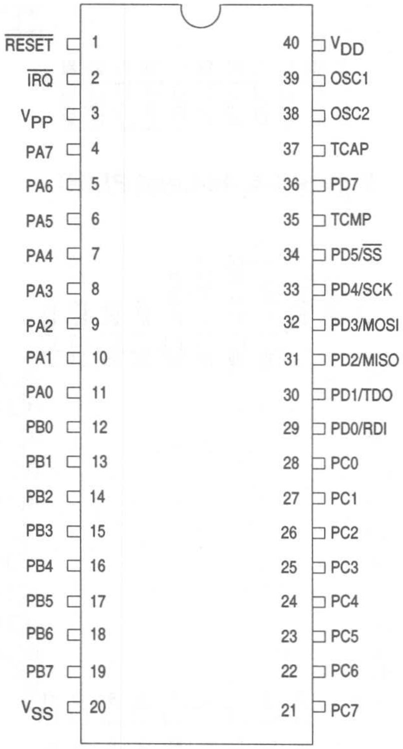


Figure 1-3. 40-Pin PDIP

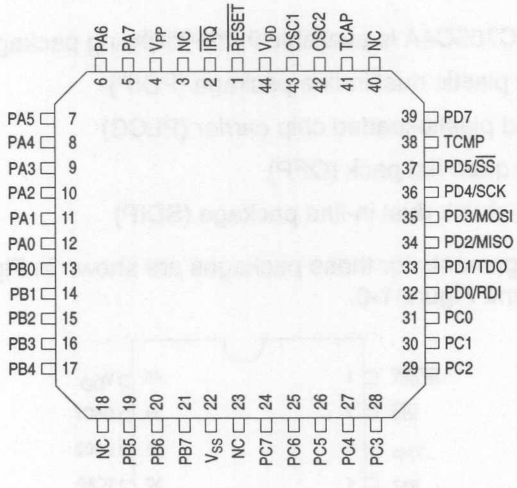


Figure 1-4. 44-Lead PLCC

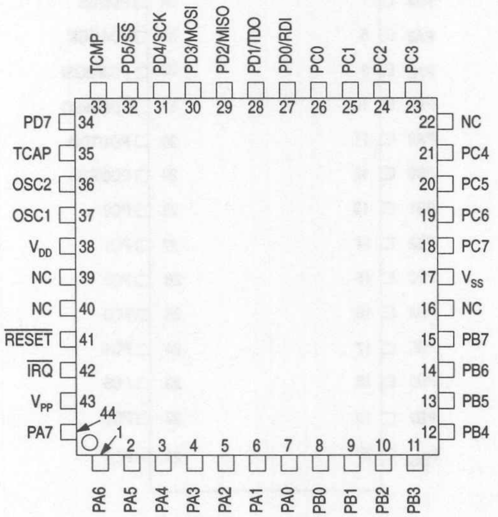


Figure 1-5. 44-Lead QFP

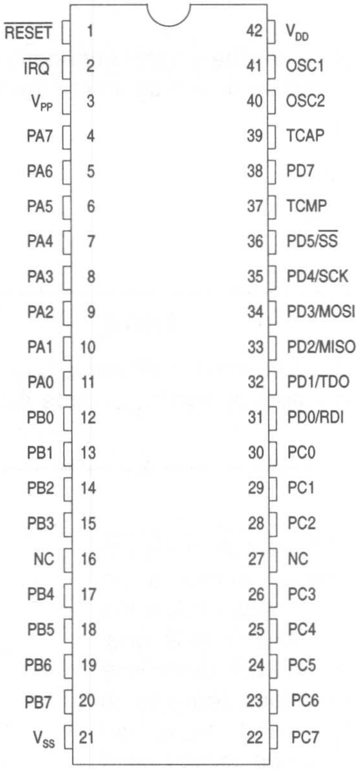


Figure 1-6. 42-Pin SDIP

1.5.1 V_{DD} and V_{SS}

The following paragraphs provide a description of the MC68HC705C4A signals. Reference is made, where applicable, to other sections that contain more detail about the function being performed.

V_{DD} and V_{SS} are the power supply and ground pins. The MCU operates from a single power supply.

Very fast signal transitions occur on the MCU pins, placing high short-duration current demands on the power supply. To prevent noise problems, take special care to provide good power supply bypassing at the MCU. Place bypass capacitors as close to the MCU as possible, as shown in Figure 1-7.

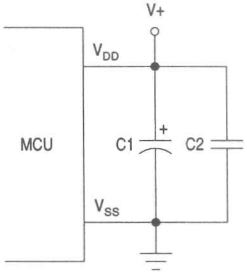


Figure 1-7. Bypassing Layout Recommendation

1.5.2 OSC1 and OSC2

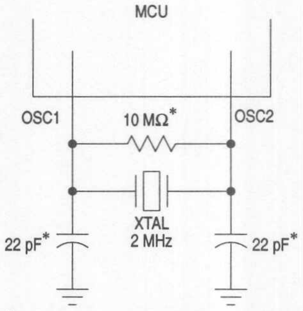
The OSC1 and OSC2 pins are the control connections for the two-pin on-chip oscillator. The oscillator can be driven by any of the following:

- Crystal resonator
- Ceramic resonator
- External clock signal

NOTE

The frequency of the internal oscillator is f_{OSC}. The MCU divides the internal oscillator output by two to produce the internal clock with a frequency of f_{OP}.

Crystal resonator — The circuit in Figure 1-8 shows a crystal oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal supplier's recommendations, because the crystal parameters determine the external component values required to provide reliable startup and maximum stability. The load capacitance values used in the oscillator circuit design should account for all stray layout capacitances. To minimize output distortion, mount the crystal and capacitors as close as possible to the pins.



*Starting value only. Follow crystal supplier's recommendations regarding component values that will provide reliable startup and maximum stability.

Figure 1-8. Crystal Connections

NOTE

Use an AT-cut crystal and not a strip or tuning fork crystal. The MCU might overdrive or have the incorrect characteristic impedance for a strip or tuning fork crystal.

Ceramic resonator — To reduce cost, use a ceramic resonator instead of a crystal. Use the circuit shown in Figure 1-9 for a two-pin ceramic resonator or the circuit shown in Figure 1-10 for a three-pin ceramic resonator, and follow the resonator manufacturer's recommendations.

The external component values required for maximum stability and reliable starting depend upon the resonator parameters. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances. To minimize output distortion, mount the resonator and capacitors as close as possible to the pins.

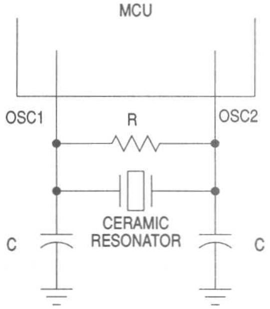


Figure 1-9. Two-Pin Ceramic Resonator Connections

NOTE

The bus frequency (f_{OP}) is one-half the external or crystal frequency (f_{OSC}), while the processor clock cycle (t_{CYC}) is two times the f_{OSC} period.

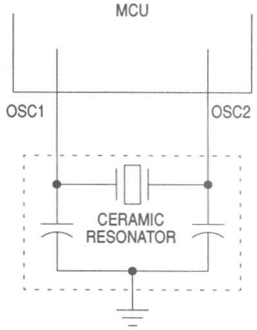


Figure 1-10. Three-Pin Ceramic Resonator Connections

External clock signal — An external clock from another CMOS-compatible device can drive the OSC1 input, with the OSC2 pin unconnected, as Figure 1-11 shows.

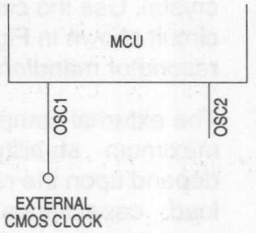


Figure 1-11. External Clock

NOTE

The bus frequency (f_{OP}) is one-half the external frequency (f_{OSC}) while the processor clock cycle is two times the f_{OSC} period.

1.5.3 External Reset Pin ($\overline{\text{RESET}}$)

A logic zero on the bidirectional $\overline{\text{RESET}}$ pin forces the MCU to a known startup state. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. (See **SECTION 5 RESETS.**)

1.5.4 External Interrupt Request Pin ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}$ pin is an asynchronous external interrupt pin. The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. (See **4.2.2 External Interrupt (IRQ)**.)

1.5.5 Input Capture Pin (TCAP)

The TCAP pin is the input capture pin for the on-chip capture/compare timer. The TCAP pin contains an internal Schmitt trigger as part of its input to improve noise immunity. (See **SECTION 8 CAPTURE/COMPARE TIMER.**)

1.5.6 Output Compare Pin (TCMP)

The TCMP pin is the output compare pin for the on-chip capture/compare timer. (See **SECTION 8 CAPTURE/COMPARE TIMER.**)

1.5.7 Port A I/O Pins (PA7–PA0)

These eight I/O lines comprise port A, a general-purpose bidirectional I/O port. The pins are programmable as either inputs or outputs under software control of the data direction registers. (See **7.2 Port A**.)

1.5.8 Port B I/O Pins (PB7–PB0)

These eight I/O pins comprise port B, a general-purpose bidirectional I/O port. The pins are programmable as either inputs or outputs under software control of the data direction registers. Port B pins also can be configured to function as external interrupts. (See **7.3 Port B**.)

1.5.9 Port C I/O Pins (PC7–PC0)

These eight I/O pins comprise port C, a general-purpose bidirectional I/O port. The pins are programmable as either inputs or outputs under software control of the data direction registers. PC7 has a high current sink and source capability (See **7.4 Port C**.)

1.5.10 Port D I/O Pins (PD7 and PD5–PD0)

These seven lines comprise port D, a fixed input port. All special functions that are enabled (SPI and SCI) affect this port. (See **7.5 Port D**.)

CAUTION

Connecting the V_{PP} pin (programming voltage) to V_{SS} (ground) could result in damage to the MCU.

SECTION 2 MEMORY

2.1 Introduction

This section describes the organization of the on-chip memory.

2.2 Memory Map

The CPU can address 8 Kbytes of memory and input/output (I/O) registers. The program counter typically advances one address at a time through memory, reading the program instructions and data. The programmable read-only memory (PROM) portion of memory—either one-time programmable read-only memory (OTPROM) or erasable programmable read-only memory (EPROM)—holds the program instructions, fixed data, user defined vectors, and interrupt service routines. The RAM portion of memory holds variable data.

I/O registers are memory-mapped so that the CPU can access their locations in the same way that it accesses all other memory locations. The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls.

Figure 2-1 is a memory map of the MCU. Addresses $0000–$001F, shown in Figure 2-2, contain most of the control, status, and data registers. Additional I/O registers have the following addresses:

- $1FDF (option register)
- $1FF0 (mask option register 1 (MOR1))
- $1FF1 (mask option register 2 (MOR2))

2.3 Input/Output (I/O)

The first 32 addresses of memory space, from $0000 to $001F, are the I/O section. These are the addresses of the I/O control registers, status registers, and data registers. See Figure 2-2 for more information.

2.4 RAM

The 176 addresses from $0050–$00FF are RAM locations. The CPU uses the top 64 RAM addresses, $00C0–00FF, as the stack. Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers. During a subroutine call, the CPU uses two bytes of the stack to store the return address. The stack pointer decrements when the CPU stores a byte on the stack and increments when the CPU retrieves a byte from the stack.

NOTE

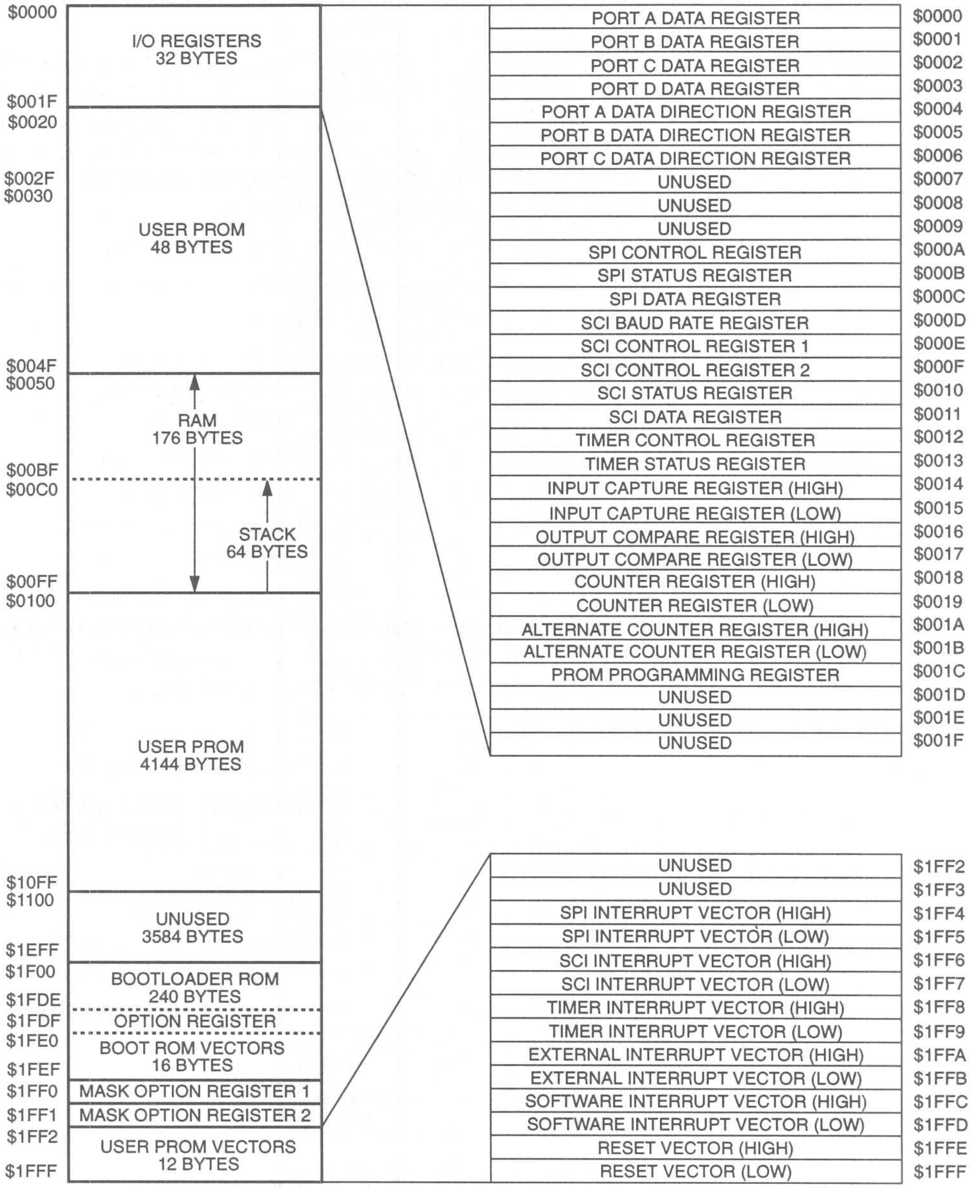
Be careful when using nested subroutines or multiple interrupt levels. The CPU can overwrite data in the stack RAM during a subroutine or during the interrupt stacking operation.

2.5 EPROM/OTPROM (PROM)

An MCU with a quartz window has a maximum of 4144 bytes of EPROM. The quartz window allows the EPROM erasure with ultraviolet light. In an MCU without a quartz window, the EPROM cannot be erased and serves a maximum 4144 bytes of OTPROM. (See **SECTION 9 EPROM/OTPROM (PROM)**.)

2.6 Bootloader ROM

The 240 bytes at addresses $1F00–$1FEF are reserved ROM addresses that contain the instructions for the bootloader functions. (See **SECTION 9 EPROM/OTPROM (PROM)**.)



PORT A DATA REGISTER	$0000
PORT B DATA REGISTER	$0001
PORT C DATA REGISTER	$0002
PORT D DATA REGISTER	$0003
PORT A DATA DIRECTION REGISTER	$0004
PORT B DATA DIRECTION REGISTER	$0005
PORT C DATA DIRECTION REGISTER	$0006
UNUSED	$0007
UNUSED	$0008
UNUSED	$0009
SPI CONTROL REGISTER	$000A
SPI STATUS REGISTER	$000B
SPI DATA REGISTER	$000C
SCI BAUD RATE REGISTER	$000D
SCI CONTROL REGISTER 1	$000E
SCI CONTROL REGISTER 2	$000F
SCI STATUS REGISTER	$0010
SCI DATA REGISTER	$0011
TIMER CONTROL REGISTER	$0012
TIMER STATUS REGISTER	$0013
INPUT CAPTURE REGISTER (HIGH)	$0014
INPUT CAPTURE REGISTER (LOW)	$0015
OUTPUT COMPARE REGISTER (HIGH)	$0016
OUTPUT COMPARE REGISTER (LOW)	$0017
COUNTER REGISTER (HIGH)	$0018
COUNTER REGISTER (LOW)	$0019
ALTERNATE COUNTER REGISTER (HIGH)	$001A
ALTERNATE COUNTER REGISTER (LOW)	$001B
PROM PROGRAMMING REGISTER	$001C
UNUSED	$001D
UNUSED	$001E
UNUSED	$001F

UNUSED	$1FF2
UNUSED	$1FF3
SPI INTERRUPT VECTOR (HIGH)	$1FF4
SPI INTERRUPT VECTOR (LOW)	$1FF5
SCI INTERRUPT VECTOR (HIGH)	$1FF6
SCI INTERRUPT VECTOR (LOW)	$1FF7
TIMER INTERRUPT VECTOR (HIGH)	$1FF8
TIMER INTERRUPT VECTOR (LOW)	$1FF9
EXTERNAL INTERRUPT VECTOR (HIGH)	$1FFA
EXTERNAL INTERRUPT VECTOR (LOW)	$1FFB
SOFTWARE INTERRUPT VECTOR (HIGH)	$1FFC
SOFTWARE INTERRUPT VECTOR (LOW)	$1FFD
RESET VECTOR (HIGH)	$1FFE
RESET VECTOR (LOW)	$1FFF

Figure 2-1. Memory Map

$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	Port A Data Register
$0001	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	Port B Data Register
$0002	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	Port C Data Register
$0003	PD7	▨	SS	SCK	MOSI	MISO	TDO	RDI	Port D Fixed Input Register
$0004	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	Port A Data Direction Register
$0005	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	Port B Data Direction Register
$0006	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	Port C Data Direction Register
$0007	▨	▨	▨	▨	▨	▨	▨	▨	Unused
$0008	▨	▨	▨	▨	▨	▨	▨	▨	Unused
$0009	▨	▨	▨	▨	▨	▨	▨	▨	Unused
$000A	SPIE	SPE	▨	MSTR	CPOL	CPHA	SPR1	SPR0	SPI Control Register (SPCR)
$000B	SPIF	WCOL	▨	MODF	▨	▨	▨	▨	SPI Status Register (SPSR)
$000C	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0	SPI Data I/O Register (SPDR)
$000D	▨	▨	SCP1	SCP0	▨	SCR2	SCR1	SCR0	SCI Baud Rate Register (BAUD)
$000E	R8	T8	▨	M	WAKE	▨	▨	▨	SCI Control Register 1 (SCCR1)
$000F	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCI Control Register 2 (SCCR2)
$0010	TDRE	TC	RDRF	IDLE	OR	NF	FE	▨	SCI Status Register (SCSR)
$0011	SCD7	SCD6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0	SCI Data Register (SCDAT)
$0012	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	Timer Control Register (TCR)
$0013	ICF	OCF	TOF	0	0	0	0	0	Timer Status Register (TSR)
$0014	Bit 15	14	13	12	11	10	9	Bit 8	Input Capture Register High (ICRH)
$0015	Bit 7	8	6	5	4	3	2	Bit 1	Input Capture Register Low (ICRL)
$0016	Bit 15	14	13	12	11	10	9	Bit 8	Output Compare Register High (OCRH)
$0017	Bit 7	8	6	5	4	3	2	Bit 1	Output Compare Register Low (OCRL)
$0018	Bit 15	14	13	12	11	10	9	Bit 8	Timer Register High (TRH)
$0019	Bit 7	8	6	5	4	3	2	Bit 1	Timer Register Low (TRL)
$001A	Bit 15	14	13	12	11	10	9	Bit 8	Alternate Timer Register High (ATRH)
$001B	Bit 7	8	6	5	4	3	2	Bit 1	Alternate Timer Register Low (ATRL)
$001C	0	0	0	0	0	LAT	0	PGM	EPROM Programming Register (PROG)
$001D	▨	▨	▨	▨	▨	▨	▨	▨	Unused
$001E	▨	▨	▨	▨	▨	▨	▨	▨	Unused
$001F	▨	▨	▨	▨	▨	▨	▨	▨	Unused

$1FDF	0	0	0	0	SEC	▨	IRQ	0	Option Register (Option)
$1FF0	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0/ COPC	Mask Option Register 1 (MOR1)
$1FF1	▨	▨	▨	▨	▨	▨	▨	NCOPE	Mask Option Register 2 (MOR2)

▨ = Unimplemented

Figure 2-2. I/O and Control Registers

SECTION 3
CENTRAL PROCESSOR UNIT (CPU)

3.1 Introduction

This section describes the central processor unit (CPU) registers.

3.2 CPU Registers

Figure 3-1 shows the five CPU registers. These are hard-wired registers within the CPU and are not part of the memory map.

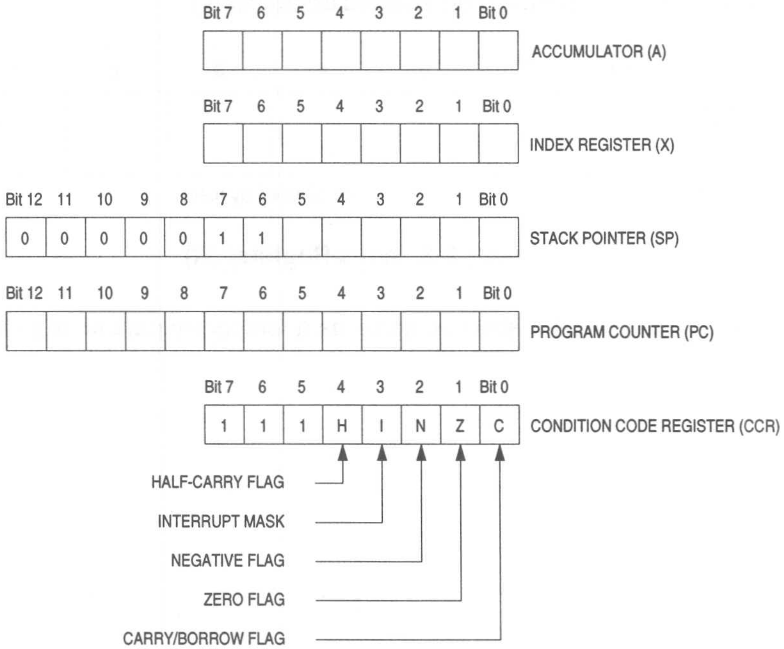


Figure 3-1. Programming Model

3.2.1 Accumulator (A)

The accumulator shown in Figure 3-2 is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and results of arithmetic and non-arithmetic operations.

		Bit 7	6	5	4	3	2	1	Bit 0
A	Read:								
	Write:								
	Reset:	Unaffected by reset							

Figure 3-2. Accumulator (A)

3.2.2 Index Register (X)

In the indexed addressing modes, the CPU uses the byte in the index register (Figure 3-3) to determine the conditional address of the operand. (See **12.2.5 Indexed, No Offset, 12.2.6 Indexed, 8-Bit Offset**, and **12.2.7 Indexed, 16-Bit Offset** for more information on indexed addressing.)

		Bit 7	6	5	4	3	2	1	Bit 0
X	Read:								
	Write:								
	Reset:	Unaffected by reset							

Figure 3-3. Index Register (X)

The 8-bit index register also can serve as a temporary data storage location.

3.2.3 Stack Pointer (SP)

The stack pointer shown in Figure 3-4 is a 13-bit register that contains the address of the next free location on the stack. During a reset or after the reset stack pointer (RSP) instruction, the stack pointer initializes to $00FF. The address in the stack pointer decrements as data is pushed onto the stack and increments as data is pulled from the stack.

		Bit 12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
SP	Read:	0	0	0	0	0	1	1						
	Write:	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	0	0	0	0	0	1	1	1	1	1	1	1	1

▨ = Unimplemented

Figure 3-4. Stack Pointer (SP)

The seven most significant bits of the stack pointer are fixed permanently at 0000011, so the stack pointer produces addresses from $00C0 to $00FF. If subroutines and interrupts use more than 64 stack locations, the stack pointer wraps around to address $00FF and begins writing over the previously stored data. A subroutine uses two stack locations. An interrupt uses five locations.

3.2.4 Program Counter (PC)

The program counter shown in Figure 3-5 is a 13-bit register that contains the address of the next instruction or operand to be fetched.

Normally, the address in the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

		Bit 12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
PC	Read: Write:													
	Reset:	Loaded with reset vector from $1FFE and $1FFF												

Figure 3-5. Program Counter (PC)

3.2.5 Condition Code Register (CCR)

The condition code register shown in Figure 3-6 is an 8-bit register whose three most significant bits are permanently fixed at 111. The condition code register contains the interrupt mask and four bits that indicate the results of prior instructions. The following paragraphs describe the functions of the condition code register.

		Bit 7	6	5	4	3	2	1	Bit 0
CCR	Read:	1	1	1	H	I	N	Z	C
	Write:	▨	▨	▨					
	Reset:	1	1	1	U	1	U	U	U

▨ = Unimplemented U = Unaffected

Figure 3-6. Condition Code Register (CCR)

Half-Carry Bit (H)

The CPU sets the half-carry flag when a carry occurs between bits 3 and 4 of the accumulator during an ADD or ADC operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations. Reset has no affect on the half-carry flag.

Interrupt Mask (I)

Setting the interrupt mask (I) disables interrupts. If an interrupt request occurs while the interrupt mask is a logic zero, the CPU saves the CPU registers on the stack, sets the interrupt mask, and then fetches the interrupt vector. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. The CPU processes the latched interrupt as soon as the interrupt mask is cleared again.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its cleared state. After a reset, the interrupt mask is set and can be cleared only by a CLI, STOP, or WAIT instruction.

Negative Flag (N)

The CPU sets the negative flag when an arithmetic operation, logical operation, or data manipulation produces a negative result (bit 7 in the results is a logic one). Reset has no effect on the negative flag.

Zero Flag (Z)

The CPU sets the zero flag when an arithmetic operation, logical operation, or data manipulation produces a result of $00. Reset has no effect on the zero flag.

Carry/Borrow Flag (C)

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some logical operations and data manipulation instructions also clear or set the carry/borrow bit. Reset has no effect on the carry/borrow flag.

3.3 Arithmetic/Logic Unit (ALU)

The ALU performs the arithmetic and logical operations defined by the instruction set. The binary arithmetic circuits decode instructions and set up the ALU for the selected operation. Most binary arithmetic is based on the addition algorithm, carrying out subtraction as negative addition. Multiplication is not performed as a discrete operation but as a chain of addition and shift operations within the ALU. The multiply instruction requires 11 internal clock cycles to complete this chain of operations.

SECTION 4
INTERRUPTS

4.1 Introduction

This section describes how interrupts temporarily change the normal processing sequence.

4.2 Interrupt Sources

The following sources can generate interrupts:

- Software instructions (SWI)
- External interrupt pin ($\overline{\text{IRQ}}$)
- Port B pins
- Capture/compare timer
  - Input capture
  - Output compare
  - Timer overflow
- SCI
  - SCI transmit data register empty
  - SCI transmission complete
  - SCI receive data register full
  - SCI receiver overrun
  - SCI receiver input idle
- SPI
  - SPI transmission complete
  - SPI mode fault
  - SPI overrun

The $\overline{\text{IRQ}}$ pin, port B pins, timer, SCI, and SPI can be masked (disabled) by setting the I bit of the CCR. The software interrupt (SWI) instruction is non-maskable.

An interrupt temporarily changes the program sequence to process a particular event. An interrupt does not stop the execution of the instruction in progress but takes effect when the current instruction completes its execution. Interrupt processing automatically saves the CPU registers on the stack and loads the program counter with a user-defined vector address.

4

4.2.1 Software Interrupt

The software interrupt instruction (SWI) causes a non-maskable interrupt.

4.2.2 External Interrupt ($\overline{\text{IRQ}}$)

An interrupt signal on the $\overline{\text{IRQ}}$ pin latches an external interrupt request. After completing the current instruction, the CPU tests the following bits:

- The IRQ latch
- The I bit in the condition code register

Setting the I bit in the condition code register disables external interrupts.

If the IRQ latch is set and the I bit is clear, the CPU then begins the interrupt sequence. The CPU clears the IRQ latch while it fetches the interrupt vector, so that another external interrupt request can be latched during the interrupt service routine. As soon as the I bit is cleared during the return from interrupt, the CPU can recognize the new interrupt request. Figure 4-1 shows the logic for external interrupts.

Figure 4-1 shows an external interrupt functional diagram. Figure 4-2 shows an external interrupt timing diagram for the interrupt line. The timing diagram illustrates two treatments of the interrupt line to the processor.

- Two single pulses on the interrupt line are spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service.

  Once a pulse occurs, the next pulse normally should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 19 instruction cycles to the total number of cycles needed to complete the service routine (not including the RTI instruction).
- Many interrupt lines are “wire-ORed” to the $\overline{\text{IRQ}}$ line. If the interrupt line remains low after servicing an interrupt, then the CPU continues to recognize an interrupt.

NOTE

The internal interrupt latch is cleared in the first part of the interrupt service routine. Therefore, a new external interrupt pulse could be latched and serviced as soon as the I bit is cleared.

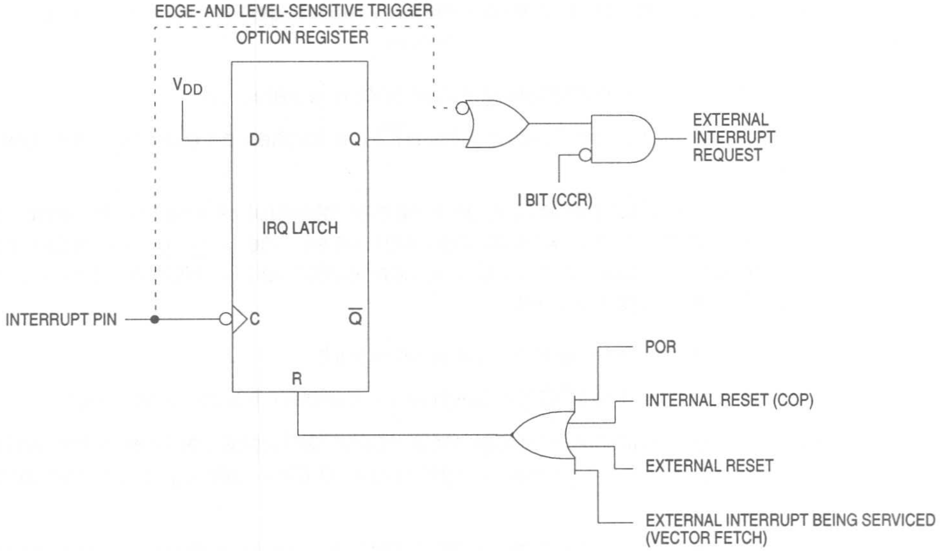
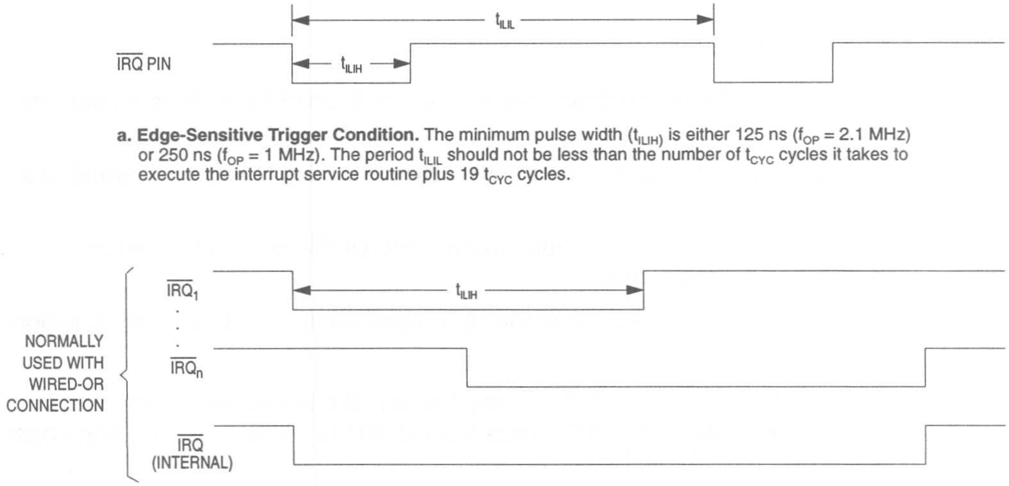


Figure 4-1. External Interrupt Internal Function Diagram



a. Edge-Sensitive Trigger Condition. The minimum pulse width (t_{ILIH}) is either 125 ns (f_{OP} = 2.1 MHz) or 250 ns (f_{OP} = 1 MHz). The period t_{ILIL} should not be less than the number of t_{CYC} cycles it takes to execute the interrupt service routine plus 19 t_{CYC} cycles.

b. Level-Sensitive Trigger Condition. If the interrupt line remains low after servicing an interrupt, then the CPU continues to recognize an interrupt.

Figure 4-2. External Interrupt Timing

The $\overline{\text{IRQ}}$ pin is negative edge-triggered only or negative edge- and low level-triggered, depending on the external interrupt triggering mask option selected.

When the edge- and level-triggering mask option is selected:

- A falling edge or a low level on the $\overline{\text{IRQ}}$ pin latches an external interrupt request.
- As long as the $\overline{\text{IRQ}}$ pin is low, an external interrupt request is present, and the CPU continues to execute the interrupt service routine. The edge- and level-sensitive trigger option allows connection to the $\overline{\text{IRQ}}$ pin of multiple wired-OR interrupt sources.

When the edge-triggering mask option is selected:

- A falling edge on the $\overline{\text{IRQ}}$ pin latches an external interrupt request.
- A subsequent external interrupt request can be latched only after the voltage level on the $\overline{\text{IRQ}}$ pin returns to logic one and then falls again to logic zero.

NOTE

If the $\overline{\text{IRQ}}$ pin is not in use, connect it to the V_{DD} pin.

4.2.3 Port B Interrupts

When the following three conditions are true, a port B pin (PBx) acts as an external interrupt pin:

- The corresponding port B pullup bit (PBPUx) in MOR1 is programmed to a logic one.
- The corresponding port B data direction bit (DDRBx) in data direction register B is a logic zero.
- The clear interrupt mask instruction (CLI) has cleared the I bit in the condition code register.

MOR1 is an EPROM register that enables the port B pullup device. (See **9.4.2 Mask Option Register 1 (MOR1)**.) Data from MOR1 is latched on the rising edge of the voltage on the $\overline{\text{RESET}}$ pin.

Port B external interrupt pins can be falling-edge sensitive only or both falling-edge and low-level sensitive, depending on the state of the IRQ bit in the option register at location $1FDF.

When the IRQ bit is a logic one, a falling edge or a low level on a port B external interrupt pin latches an external interrupt request. As long as any port B external interrupt pin is low, an external interrupt request is present, and the CPU continues to execute the interrupt service routine.

When the IRQ bit is a logic zero, a falling-edge only on a port B external interrupt pin latches an external interrupt request. A subsequent port B external interrupt request can be latched only after the voltage level of the previous port B external interrupt signal returns to a logic one and then falls again to a logic zero.

Figure 4-3 shows the port B I/O logic.

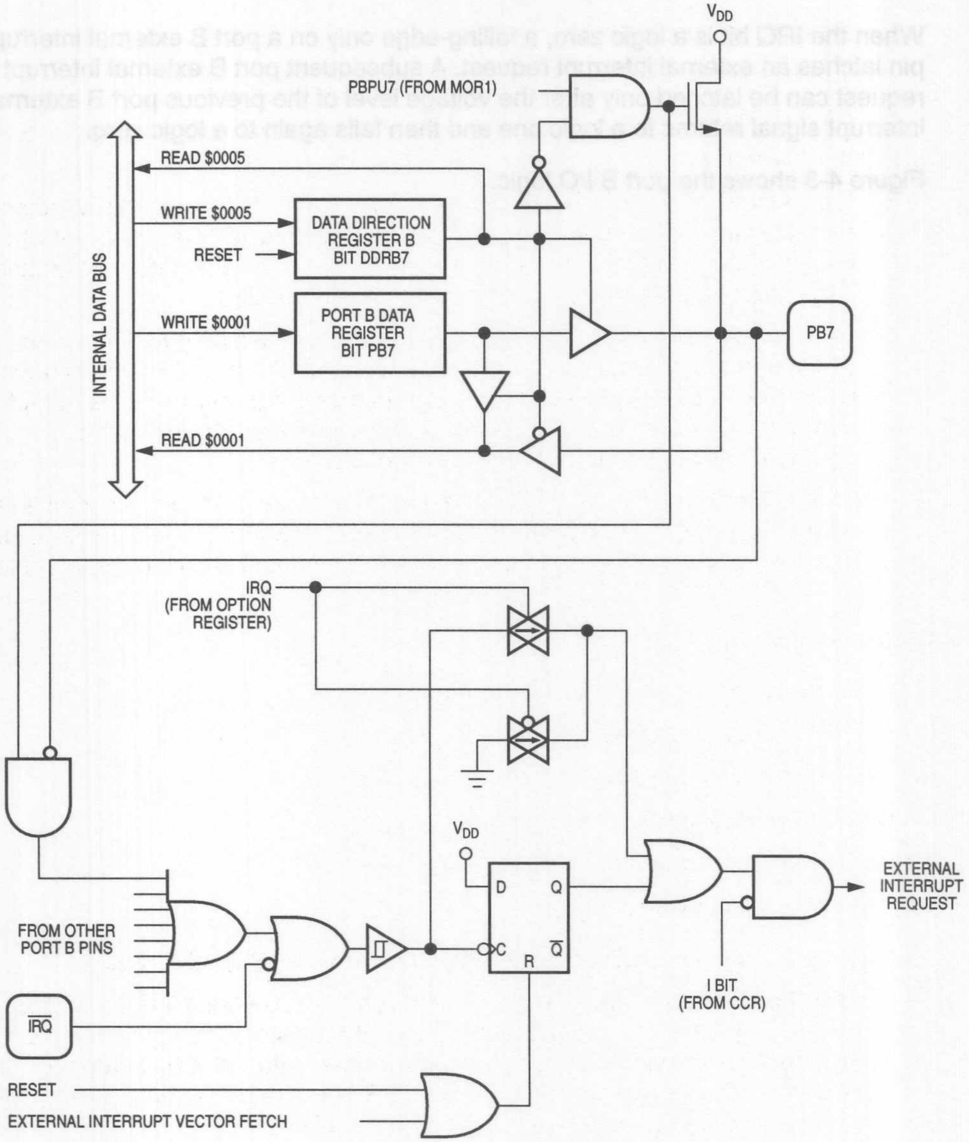

VDD
PBPU7 (FROM MOR1)
READ $0005
WRITE $0005
RESET
DATA DIRECTION REGISTER B BIT DDRB7
INTERNAL DATA BUS
WRITE $0001
PORT B DATA REGISTER BIT PB7
PB7
READ $0001
IRQ (FROM OPTION REGISTER)
VDD
D
Q
C
R
Q̄
EXTERNAL INTERRUPT REQUEST
FROM OTHER PORT B PINS
IRQ
I BIT (FROM CCR)
RESET
EXTERNAL INTERRUPT VECTOR FETCH

Figure 4-3. Port B I/O Logic

4.2.4 Capture/Compare Timer Interrupts

The capture/compare timer can generate the following interrupts:

- Input capture interrupt
- Output compare interrupt
- Timer overflow interrupt

Setting the I bit in the condition code register disables all interrupts except for SWI and timer interrupts.

- Input Capture Interrupt — The input capture flag bit (ICF) indicates a transfer from the timer registers to the input capture registers. ICF becomes set when an active edge occurs on the TCAP pin. ICF generates an interrupt request if the input capture interrupt enable bit (ICIE) is set also.
- Output Compare Interrupt — The output compare flag bit (OCF) indicates a transfer of the output level bit (OLVL) to the TCMP pin. OCF becomes set when the 16-bit counter counts up to the value programmed in the output compare registers. OCF generates an interrupt request if the output compare interrupt enable bit (OCIE) is set also.
- Timer Overflow Interrupt — The timer overflow flag bit (TOF) becomes set when the 16-bit counter rolls over from $FFFF to $0000. TOF generates an interrupt request if the timer overflow interrupt enable bit (TOIE) is set also.

4.2.5 SCI Interrupts

The SCI can generate the following interrupts:

- Transmit data register empty interrupt
- Transmission complete interrupt
- Receive data register full interrupt
- Receiver overrun interrupt
- Receiver input idle interrupt

Setting the I bit in the condition code register disables all SCI interrupts.

- SCI Transmit Data Register Empty Interrupt — The transmit data register empty bit (TDRE) indicates that the SCI data register is ready to receive a byte for transmission. TDRE becomes set when data in the SCI data register transfers to the transmit shift register. TDRE generates an interrupt request if the transmit interrupt enable bit (TIE) is set also.

- SCI Transmission Complete Interrupt — The transmission complete bit (TC) indicates the completion of an SCI transmission. TC becomes set when the TDRE bit becomes set and no data, preamble, or break character is being transmitted. TC generates an interrupt request if the transmission complete interrupt enable bit (TCIE) is set also.
- SCI Receive Data Register Full Interrupt — The receive data register full bit (RDRF) indicates that a byte is ready to be read in the SCI data register. RDRF becomes set when the data in the receive shift register transfers to the SCI data register. RDRF generates an interrupt request if the receive interrupt enable bit (RIE) is set also.
- SCI Receiver Overrun Interrupt — The overrun bit (OR) indicates that a received byte is lost because software has not read the previously received byte. OR becomes set when a byte shifts into the receive shift register before software reads the word already in the SCI data register. OR generates an interrupt request if the receive interrupt enable bit (RIE) is set also.
- SCI Receiver Input Idle Interrupt — The receiver input idle bit (IDLE) indicates that the SCI receiver input is not receiving data. IDLE becomes set when 10 or 11 consecutive logic ones appear on the receiver input. IDLE generates an interrupt request if the idle line interrupt enable bit (ILIE) is set also.

4.2.6 SPI Interrupts

The SPI can generate the following interrupts:

- SPI transmission complete interrupt
- SPI mode fault interrupt

Setting the I bit in the condition code register disables all SPI interrupts.

- SPI Transmission Complete Interrupt — The SPI flag bit (SPIF) in the SPI status register indicates the completion of an SPI transmission. SPIF becomes set when a byte shifts into or out of the SPI data register. SPIF generates an interrupt request if the SPIE bit is set also.
- SPI Mode Fault Interrupt — The mode fault bit (MODF) in the SPI status register indicates an SPI mode error. MODF becomes set when a logic zero occurs on the PD5/$\overline{\text{SS}}$ pin while the master bit (MSTR) in the SPI control register is set. MODF generates an interrupt request if the SPIE bit is set also.

4.3 Interrupt Processing

The CPU takes the following actions to begin servicing an interrupt:

1. Stores the CPU registers on the stack in the order shown in **Figure 4-4. Interrupt Stacking Order**
2. Sets the I bit in the condition code register to prevent further interrupts
3. Loads the program counter with the contents of the appropriate interrupt vector locations as shown in Table 4-1.

Table 4-1. Reset/Interrupt Vector Addresses

Function	Source	Local Mask	Global Mask	Priority (1 = Highest)	Vector Address
Reset	Power-On Logic	None	None	1	$1FFE–$1FFF
	$\overline{\text{RESET}}$ Pin				
Software Interrupt (SWI)	User Code	None	None	Same Priority As Any Instruction	$1FFC–$1FFD
External Interrupt	$\overline{\text{IRQ}}$ Pin	None	I Bit	2	$1FFA–$1FFB
	Port B Pins				
Timer Interrupts	ICF Bit	ICIE Bit	I Bit	3	$1FF8–$1FF9
	OCF Bit	OCIE Bit			
	TOF Bit	TOIE Bit			
SCI Interrupts	TDRE Bit	TCIE Bit	I Bit	4	$1FF6–$1FF7
	TC Bit				
	RDRF Bit	RIE Bit			
	OR Bit				
	IDLE Bit	ILIE Bit			
SPI Interrupts	SPIF Bit	SPIE	I Bit	5	$1FF4–$1FF5
	MODF Bit				

The return from interrupt (RTI) instruction causes the CPU to recover the CPU registers from the stack as shown in **Figure 4-4. Interrupt Stacking Order**.

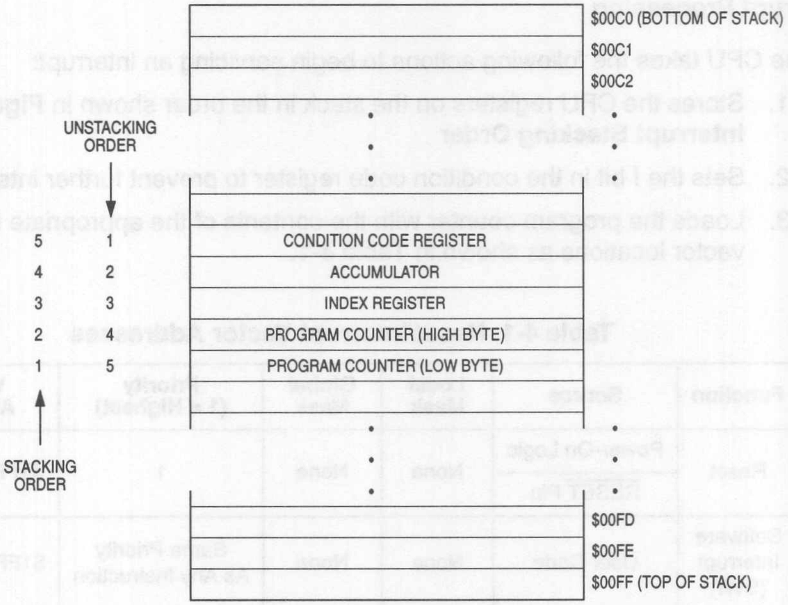


Figure 4-4. Interrupt Stacking Order

NOTE

If more than one interrupt request is pending, the CPU fetches the vector of the higher priority interrupt first. A higher priority interrupt does not interrupt a lower priority interrupt service routine unless the lower priority interrupt service routine clears the I bit. See **Table 4-1. Reset/Interrupt Vector Addresses** for a priority listing.

Figure 4-5 shows the sequence of events caused by an interrupt.

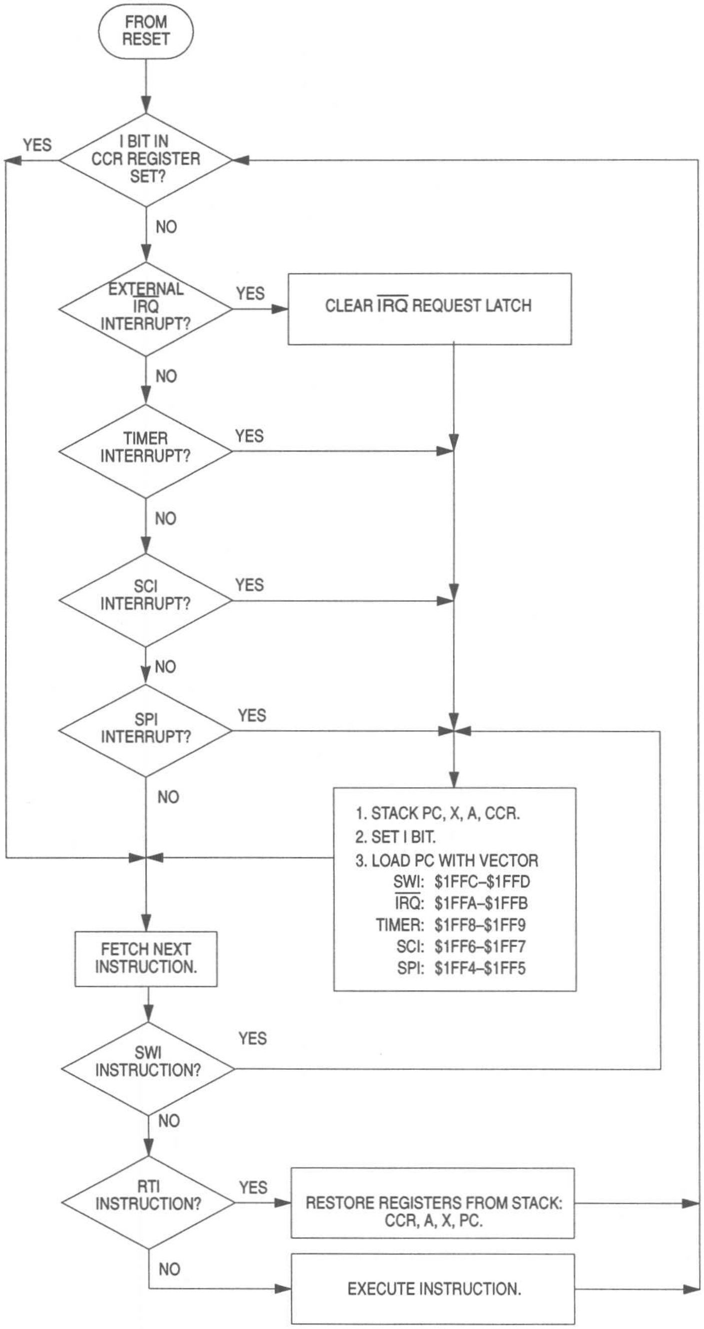


Figure 4-5. Reset and Interrupt Processing Flowchart

SECTION 5
RESETS

5.1 Introduction

This section describes how resets initialize the MCU.

5.2 Reset Sources

A reset immediately stops the operation of the instruction being executed, initializes certain control bits, and loads the program counter with a user-defined reset vector address. The following conditions produce a reset:

- Power-on reset — Initial power-up
- External reset — A logic zero applied to the $\overline{\text{RESET}}$ pin
- Internal computer operating properly (COP) watchdog timer reset

5.2.1 Power-On Reset (POR)

A positive transition on the V_{DD} pin generates a power-on reset. The power-on reset is strictly for the power-up condition and cannot be used to detect drops in power supply voltage.

A 4064 t_{CYC} (internal clock cycle) delay after the oscillator becomes active allows the clock generator to stabilize. If the $\overline{\text{RESET}}$ pin is at logic zero at the end of 4064 t_{CYC}, the MCU remains in the reset condition until the signal on the $\overline{\text{RESET}}$ pin goes to logic one.

5.2.2 External Reset

The minimum time required for the MCU to recognize a reset is 1 1/2 t_{CYC}. However, to guarantee that the MCU recognizes an external reset as an external reset and not as a COP watchdog timer reset, the $\overline{\text{RESET}}$ pin must be low for 8 t_{CYC}. After 6 t_{CYC}, the input on the $\overline{\text{RESET}}$ pin is sampled. If the pin is still low, an external reset has occurred. If the input is high, then the MCU assumes that the reset was initiated internally by the COP watchdog timer. This method of differentiating between external and internal reset conditions assumes that the $\overline{\text{RESET}}$ pin will rise to a logic one less than 2 t_{CYC} after its release and that an externally generated reset should stay active for at least 8 t_{CYC}.

5.2.3 COP Watchdog Reset

A timeout of the COP watchdog timer generates a COP reset. A COP watchdog timer, once enabled, is part of a software error detection system and must be cleared periodically to start a new timeout period. (For information on the COP watchdog timer in low-power modes, refer to **SECTION 6 LOW-POWER MODES**.)

A timeout of the 18-stage ripple counter in the COP watchdog generates a reset. The timeout period is 64 ms when f_{OSC} = 4 MHz. Two memory locations control operation of the COP watchdog:

- COP enable bit (NCOPE) in mask option register 2 (MOR2) (See **9.4.3 Mask Option Register 2 (MOR2)**.) — Programming the NCOPE bit in MOR2 to a logic one enables the COP watchdog.
- COP clear bit (COPC) at address \$1FF0 — To clear the COP watchdog and start a new COP timeout period, write a logic zero to bit 0 of address \$1FF0. Reading address \$1FF0 returns the mask option register 1 (MOR1) data at that location. (See **9.4.2 Mask Option Register 1 (MOR1)**.)

Figure 5-1 is a diagram of the COP.

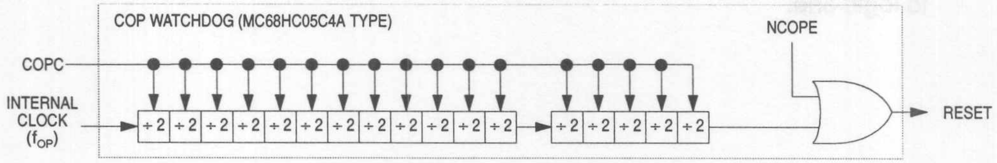


Figure 5-1. COP Watchdog Diagram

SECTION 6
LOW-POWER MODES

6.1 Introduction

This section describes the three low-power modes:

- Stop mode
- Wait mode
- Data-retention mode

6.2 Stop Mode

The STOP instruction places the MCU in its lowest power consumption mode. In the stop mode, the internal oscillator is turned off, halting all internal processing including timer, SCI, and master mode SPI operation. (See **Figure 6-1. Stop/Wait Mode Function Flowchart**.)

During the stop mode, the I bit in the condition code register (CCR) is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output (I/O) lines remain unchanged. The processor can be brought out of the stop mode only by an external interrupt or reset.

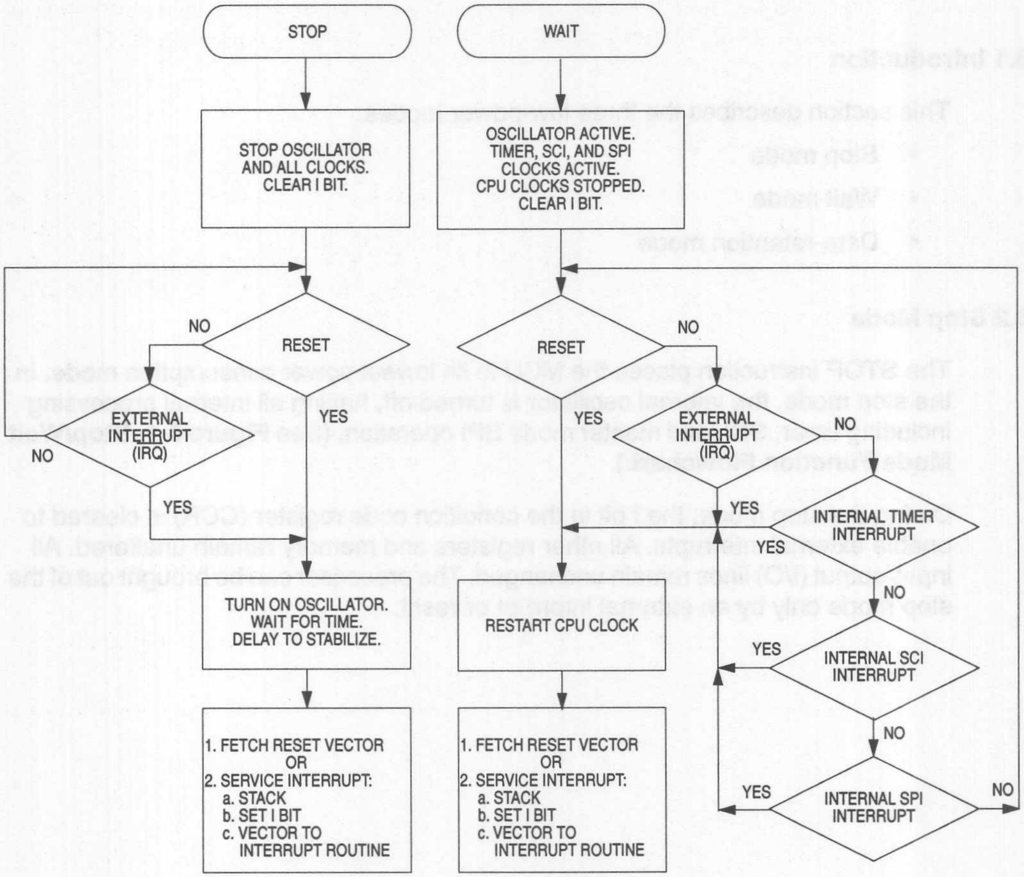

STOP
WAIT
STOP OSCILLATOR AND ALL CLOCKS. CLEAR I BIT.
OSCILLATOR ACTIVE. TIMER, SCI, AND SPI CLOCKS ACTIVE. CPU CLOCKS STOPPED. CLEAR I BIT.
RESET
NO
YES
EXTERNAL INTERRUPT (IRQ)
NO
YES
RESET
NO
YES
EXTERNAL INTERRUPT (IRQ)
NO
YES
INTERNAL TIMER INTERRUPT
YES
NO
INTERNAL SCI INTERRUPT
YES
NO
INTERNAL SPI INTERRUPT
YES
NO
TURN ON OSCILLATOR. WAIT FOR TIME. DELAY TO STABILIZE.
RESTART CPU CLOCK
1. FETCH RESET VECTOR OR 2. SERVICE INTERRUPT: a. STACK b. SET I BIT c. VECTOR TO INTERRUPT ROUTINE
1. FETCH RESET VECTOR OR 2. SERVICE INTERRUPT: a. STACK b. SET I BIT c. VECTOR TO INTERRUPT ROUTINE

Figure 6-1. Stop/Wait Mode Function Flowchart

6.2.1 SCI During Stop Mode

When the MCU enters the stop mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the $\overline{\text{IRQ}}$ pin is used to exit stop mode, the transfer resumes.

If the SCI receiver is receiving data and the stop mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. Therefore, all SCI transfers should be in the idle state when the STOP instruction is executed.

6.2.2 SPI During Stop Mode

When the MCU enters the stop mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits the stop mode by a low signal on the $\overline{\text{IRQ}}$ pin. If reset is used to exit the stop mode, the SPI control and status bits are cleared, and the SPI is disabled.

If the MCU is in slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device. At the end of a possible transmission with a slave SPI in the stop mode, no flags are set until a low on the $\overline{\text{IRQ}}$ pin wakes up the MCU.

NOTE

Although a slave SPI in stop mode can exchange data with a master SPI, the status bits of a slave SPI are inactive in stop mode.

6.2.3 COP Watchdog in Stop Mode

The STOP instruction has the following effects on the COP watchdog:

- Turns off the oscillator and turns off the COP watchdog counter
- Clears the COP watchdog counter

If the $\overline{\text{RESET}}$ pin brings the MCU out of stop mode, the COP watchdog begins counting immediately. The reset function clears the COP counter again after the 4064-t_{CYC} clock stabilization delay.

If the $\overline{\text{IRQ}}$ pin brings the MCU out of stop mode, the COP watchdog begins counting immediately. The IRQ function does not clear the COP counter again after the 4064-t_{CYC} clock stabilization delay. (See Figure 6-2.)

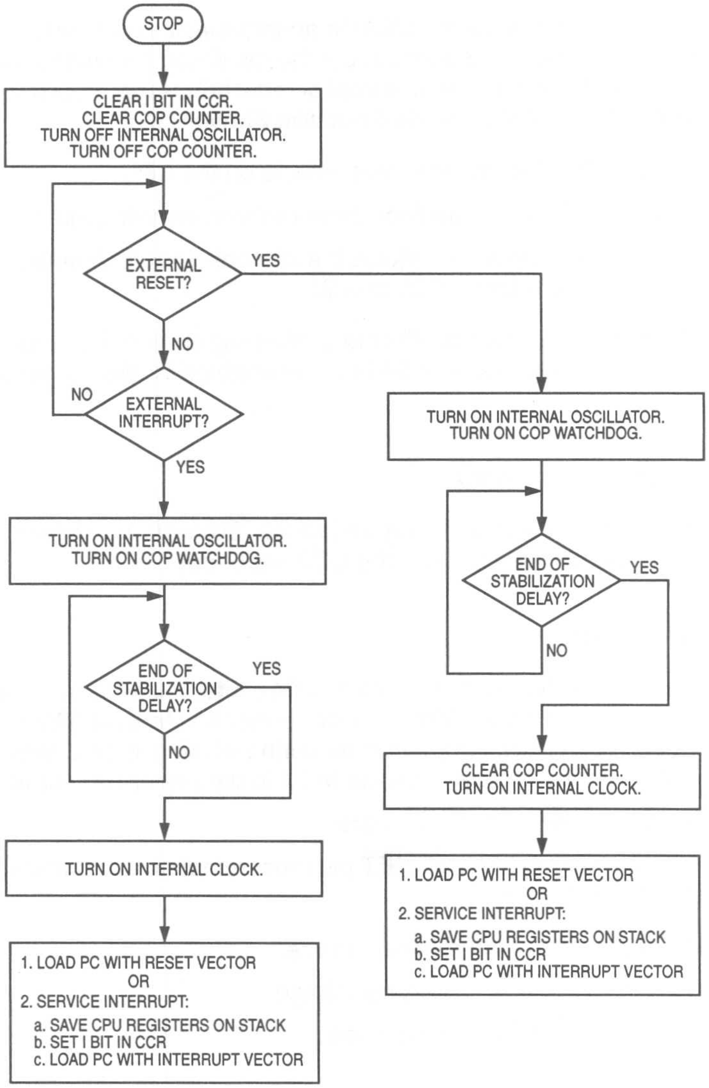

STOP
CLEAR I BIT IN CCR.
CLEAR COP COUNTER.
TURN OFF INTERNAL OSCILLATOR.
TURN OFF COP COUNTER.
EXTERNAL RESET?
YES
NO
EXTERNAL INTERRUPT?
NO
YES
TURN ON INTERNAL OSCILLATOR.
TURN ON COP WATCHDOG.
END OF STABILIZATION DELAY?
YES
NO
TURN ON INTERNAL CLOCK.
1. LOAD PC WITH RESET VECTOR
OR
2. SERVICE INTERRUPT:
a. SAVE CPU REGISTERS ON STACK
b. SET I BIT IN CCR
c. LOAD PC WITH INTERRUPT VECTOR
TURN ON INTERNAL OSCILLATOR.
TURN ON COP WATCHDOG.
END OF STABILIZATION DELAY?
YES
NO
CLEAR COP COUNTER.
TURN ON INTERNAL CLOCK.
1. LOAD PC WITH RESET VECTOR
OR
2. SERVICE INTERRUPT:
a. SAVE CPU REGISTERS ON STACK
b. SET I BIT IN CCR
c. LOAD PC WITH INTERRUPT VECTOR

Figure 6-2. COP Watchdog in Stop Mode Flowchart (NCOPE = 1)

6.3 Wait Mode

The WAIT instruction places the MCU in an intermediate power-consumption mode. All CPU activity is suspended, but the oscillator, capture/compare timer, SCI, and SPI remain active. Any interrupt or reset brings the MCU out of wait mode. (See **Figure 6-1. Stop/Wait Mode Function Flowchart.**)

The WAIT instruction has the following effects on the CPU:

- Clears the I bit in the condition code register, enabling interrupts
- Stops the CPU clock, but allows the internal clock to drive the capture/compare timer, SCI, and SPI

The WAIT instruction does not affect any other registers or I/O lines. The capture/compare timer, SCI, and SPI can be enabled to allow a periodic exit from wait mode.

6.3.1 COP Watchdog in Wait Mode

The COP watchdog is active during wait mode. Software must periodically bring the MCU out of wait mode to clear the COP watchdog.

6.4 Data-Retention Mode

In data-retention mode, the MCU retains RAM contents and CPU register contents at V_{DD} voltages as low as 2.0 Vdc. The data-retention feature allows the MCU to remain in a low-power-consumption state during which it retains data, but the CPU cannot execute instructions. To put the MCU in data-retention mode:

1. Drive the $\overline{\text{RESET}}$ pin to logic zero.
2. Lower V_{DD} voltage. The $\overline{\text{RESET}}$ pin must remain low continuously during data-retention mode.

To take the MCU out of data-retention mode:

1. Return V_{DD} to normal operating voltage.
2. Return the $\overline{\text{RESET}}$ pin to logic one.

SECTION 7
PARALLEL INPUT/OUTPUT (I/O)

7.1 Introduction

This section describes the programming of ports A, B, C, and D.

7.2 Port A

Port A is an 8-bit general-purpose bidirectional I/O port.

7.2.1 Port A Data Register (PORTA)

The port A data register (Figure 7-1) contains a data latch for each of the eight port A pins. When a port A pin is programmed to be an output, the state of its data register bit determines the state of the output pin. When a port A pin is programmed to be an input, reading the port A data register returns the logic state of the pin.

		Bit 7	6	5	4	3	2	1	Bit 0
PORTA $0000	Read: Write:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	Reset:	Unaffected by reset							

Figure 7-1. Port A Data Register (PORTA)

PA7–PA0 — Port A Data Bits

These read/write bits are software programmable. Data direction of each bit is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

7.2.2 Data Direction Register A (DDRA)

The contents of data direction register A shown in Figure 7-2 determine whether each port A pin is an input or an output. Writing a logic one to a DDRA bit enables the output buffer for the associated port A pin; a logic zero disables the output buffer. A reset clears all DDRA bits, configuring all port A pins as inputs.

		Bit 7	6	5	4	3	2	1	Bit 0
DDRA $0004	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
	Reset:	0	0	0	0	0	0	0	0

Figure 7-2. Data Direction Register A (DDRA)

DDRA7–DDRA0 — Port A Data Direction Bits

These read/write bits control port A data direction. Reset clears bits DDRA7–DDRA0.

1 = Corresponding port A pin configured as output
0 = Corresponding port A pin configured as input

NOTE

Avoid glitches on port A pins by writing to the port A data register before changing DDRA bits from logic zero to logic one.

7.2.3 Port A Logic

Figure 7-3 is a diagram of the port A I/O logic.

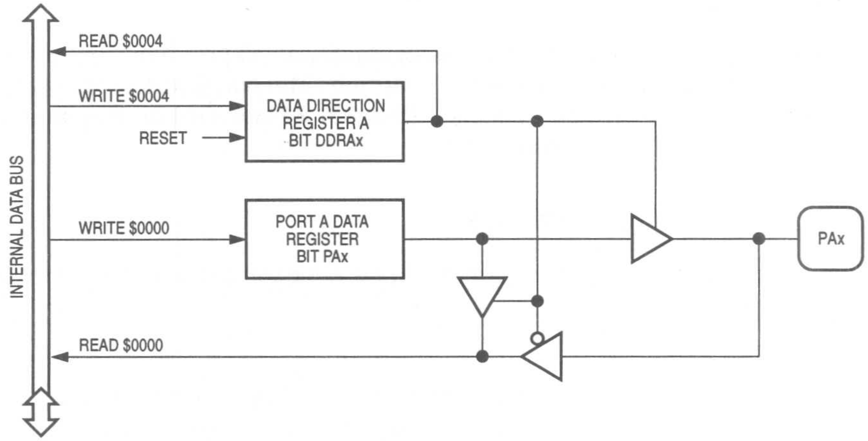


Figure 7-3. Port A I/O Logic

When a port A pin is programmed to be an output, the state of its data register bit determines the state of the output pin. When a port A pin is programmed to be an input, reading the port A data register returns the logic state of the pin.The data latch can always be written, regardless of the state of its DDRA bit. Table 7-1 summarizes the operation of the port A pins.

Table 7-1. Port A Pin Functions

DDRA Bit	I/O Pin Mode	Accesses to DDRA	Accesses to PORTA	
		Read/Write	Read	Write
0	Input, Hi-Z[1]	DDRA7–DDRA0	Pin	PA7–PA0[2]
1	Output	DDRA7–DDRA0	PA7–PA0	PA7–PA0

NOTES:
1. Hi-Z = high impedance
2. Writing affects data register but does not affect input.

NOTE

To avoid excessive current draw, tie all unused input pins to V_{DD} or V_{SS}, or change I/O pins to outputs by writing to DDRA in user code as early as possible.

7.3 Port B

Port B is an 8-bit general-purpose bidirectional I/O port. Port B pins can also be configured to function as external interrupts. The port B pullup devices are enabled in mask option register 1 (MOR1). (See **9.4.2 Mask Option Register 1 (MOR1)** and **4.2.3 Port B Interrupts.**)

7.3.1 Port B Data Register (PORTB)

The port B data register shown in Figure 7-4 contains a data latch for each of the eight port B pins.

		Bit 7	6	5	4	3	2	1	0
PORTB $0001	Read: Write:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
	Reset:	Unaffected by reset							

Figure 7-4. Port B Data Register (PORTB)

PB7–PB0 — Port B Data Bits

These read/write bits are software programmable. Data direction of each bit is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

7.3.2 Data Direction Register B (DDRB)

The contents of data direction register B shown in Figure 7-5 determine whether each port B pin is an input or an output. Writing a logic one to a DDRB bit enables the output buffer for the associated port B pin; a logic zero disables the output buffer. A reset clears all DDRB bits, configuring all port B pins as inputs. If the pullup devices are enabled by mask option, setting a DDRB bit to a logic one turns off the pullup device for that pin.

		Bit 7	6	5	4	3	2	1	Bit 0
DDRB $0005	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
	Reset:	0	0	0	0	0	0	0	0

Figure 7-5. Data Direction Register B (DDRB)

DDRB7–DDRB0 — Port B Data Direction Bits

These read/write bits control port B data direction. Reset clears bits DDRB7–DDRB0.

1 = Corresponding port B pin configured as output

0 = Corresponding port B pin configured as input

NOTE

Avoid glitches on port B pins by writing to the port B data register before changing DDRB bits from logic zero to logic one.

7.3.3 Port B Logic

Figure 7-6 shows the port B I/O logic.

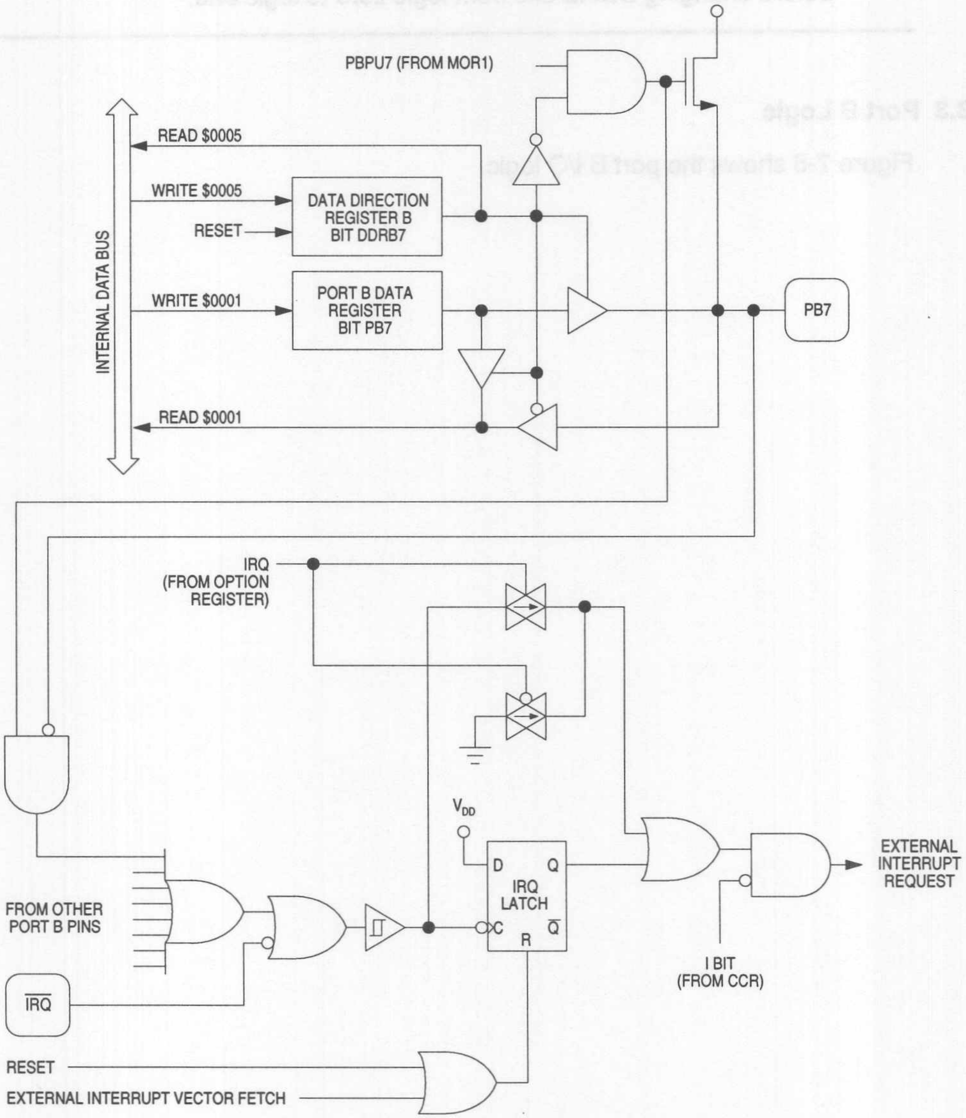


Figure 7-6. Port B I/O Logic

When a port B pin is programmed as an output, reading the port bit reads the value of the data latch and not the voltage on the pin itself. When a port B pin is programmed as an input, reading the port bit reads the voltage level on the pin. The data latch can always be written, regardless of the state of its DDRB bit.

Table 7-2. Port B Pin Functions

DDRB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PORTB	
		Read/Write	Read	Write
0	Input, Hi-Z[(1)]	DDRB7–DDRB0	Pin	PB7–PB0[(2)]
1	Output	DDRB7–DDRB0	PB7–PB0	PB7–PB0

NOTES:
1. Hi-Z = high impedance
2. Writing affects data register but does not affect input.

NOTE

To avoid excessive current draw, tie all unused input pins to V_{DD} or V_{SS}, or for I/O pins change to outputs by writing to DDRB in user code as early as possible.

7.4 Port C

Port C is an 8-bit general-purpose bidirectional I/O port. PC7 has a high current sink and source capability.

7.4.1 Port C Data Register (PORTC)

The port C data register shown in Figure 7-7 contains a data latch for each of the eight port C pins. When a port C pin is programmed to be an output, the state of its data register bit determines the state of the output pin. When a port C pin is programmed to be an input, reading the port C data register returns the logic state of the pin.

PORTC $0002		Bit 7	6	5	4	3	2	1	Bit 0
	Read: Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
	Reset:	Unaffected by reset							

Figure 7-7. Port C Data Register (PORTC)

PC7–PC0 — Port C Data Bits

These read/write bits are software programmable. Data direction of each bit is under the control of the corresponding bit in data direction register C. PC7 has a high current sink and source capability. Reset has no effect on port C data.

7.4.2 Data Direction Register C (DDRC)

The contents of data direction register C shown in Figure 7-8 determine whether each port C pin is an input or an output. Writing a logic one to a DDRC bit enables the output buffer for the associated port C pin; a logic zero disables the output buffer. A reset clears all DDRC bits, configuring all port C pins as inputs.

		Bit 7	6	5	4	3	2	1	Bit 0
DDRC $0006	Read: Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
	Reset:	0	0	0	0	0	0	0	0

Figure 7-8. Data Direction Register C (DDRC)

DDRC7–DDRC0 — Port C Data Direction Bits

These read/write bits control port C data direction. Reset clears bits DDRC7–DDRC0.

1 = Corresponding port C pin configured as output
0 = Corresponding port C pin configured as input

NOTE

Avoid glitches on port C pins by writing to the port C data register before changing DDRC bits from logic zero to logic one.

7.4.3 Port C Logic

Figure 7-9 shows port C I/O logic.

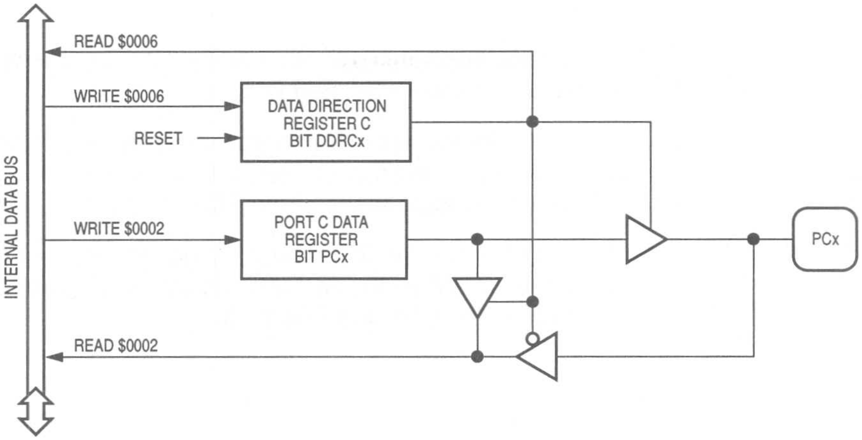


Figure 7-9. Port C I/O Logic

When a port C pin is programmed as an output, reading the port bit reads the value of the data latch and not the voltage on the pin. When a port C pin is programmed as an input, reading the port bit reads the voltage level on the pin. The data latch can always be written, regardless of the state of its DDRC bit. Table 7-1 summarizes the operation of the port C pins.

Table 7-3. Port C Pin Functions

DDRC Bit	I/O Pin Mode	Accesses to DDRC	Accesses to PORTC	
		Read/Write	Read	Write
0	Input, Hi-Z(1)	DDRC7–DDRC0	Pin	PC7–PC0(2)
1	Output	DDRC7–DDRC0	PC7–PC0	PC7–PC0

NOTES:
1. Hi-Z = high impedance
2. Writing affects data register but does not affect input.

NOTE

To avoid excessive current draw, tie all unused input pins to V_{DD} or V_{SS}, or change I/O pins to outputs by writing to DDRC in user code as early as possible.

7.5 Port D

Port D is a 7-bit special-purpose input-only port that has no data register. Reading address $0003 returns the logic states of the port D pins.

Port D shares pins PD5–PD2 with the serial peripheral interface module (SPI). When the SPI is enabled, PD5–PD2 read as logic zeros. When the SPI is disabled, reading address $0003 returns the logic states of the PD5–PD2 pins.

Port D shares pins PD1 and PD0 with the SCI module. When the SCI is enabled, PD1 and PD0 read as logic zeros. When the SCI is disabled, reading address $0003 returns the logic states of the PD1 and PD0 pins.

SECTION 8
CAPTURE/COMPARE TIMER

8.1 Introduction

This section describes the operation of the 16-bit capture/compare timer. Figure 8-1 shows the structure of the timer module. Figure 8-2 is a summary of the timer I/O registers.

8.2 Timer Operation

The core of the capture/compare timer is a 16-bit free-running counter. The counter is the timing reference for the input capture and output compare functions. The input capture and output compare functions can latch the times at which external events occur, measure input waveforms, and generate output waveforms and timing delays. Software can read the value in the counter at any time without affecting the counter sequence.

Because of the 16-bit timer architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers.

Because the counter is 16 bits long and preceded by a fixed divide-by-four prescaler, the counter rolls over every 262,144 internal clock cycles. Timer resolution with a 4-MHz crystal is 2 µs.

8.2.1 Input Capture

The input capture function can record the time at which an external event occurs. When the input capture circuitry detects an active edge on the input capture pin (TCAP), it latches the contents of the timer registers into the input capture registers. The polarity of the active edge is programmable.

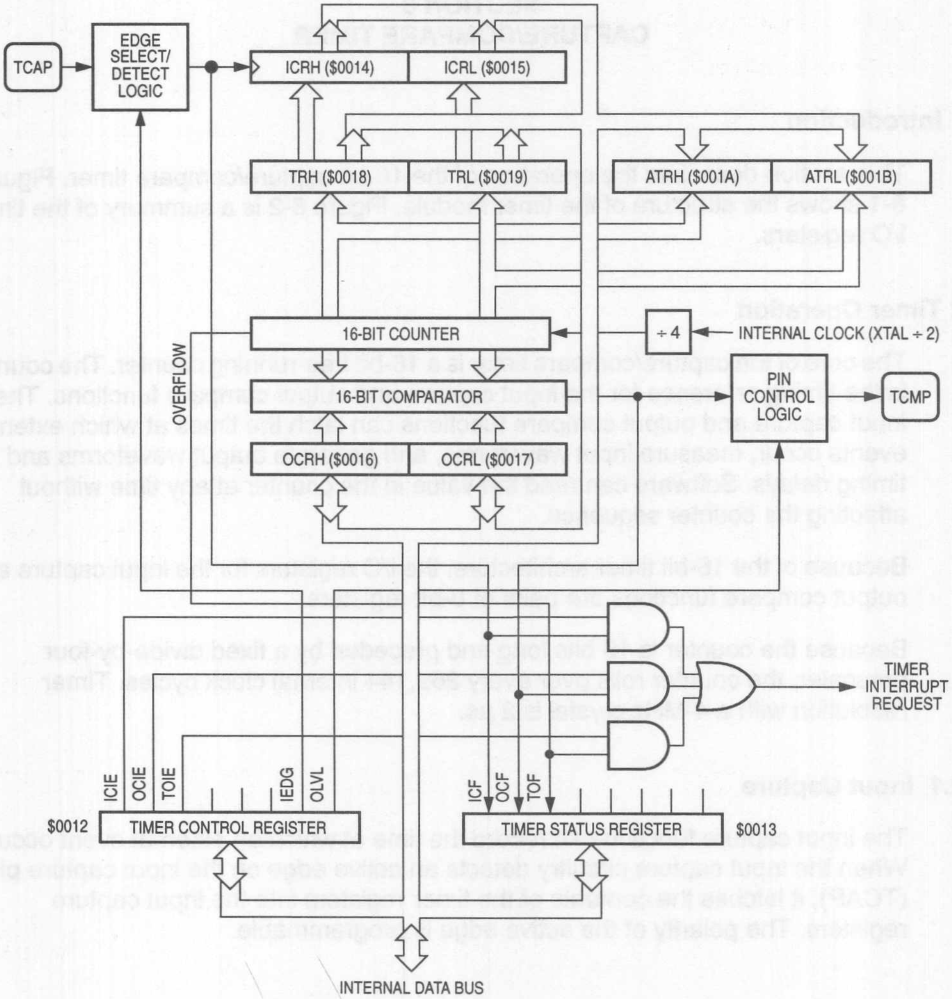


Figure 8-1. Timer Block Diagram

Register Name	Bit 7	6	5	4	3	2	1	Bit 0	Addr.
Timer Control Register (TCR)	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	$0012
Timer Status Register (TSR)	ICF	OCF	TOF	0	0	0	0	Bit 0	$0013
Input Capture Register High (ICRH)	Bit 7	6	5	4	3	2	1	Bit 0	$0014
Input Capture Register Low (ICRL)	Bit 7	6	5	4	3	2	1	Bit 0	$0015
Output Compare Register High (OCRH)	Bit 7	6	5	4	3	2	1	Bit 0	$0016
Output Compare Register Low (OCRL)	Bit 7	6	5	4	3	2	1	Bit 0	$0017
Timer Register High (TRH)	Bit 7	6	5	4	3	2	1	Bit 0	$0018
Timer Register Low (TRL)	Bit 7	6	5	4	3	2	1	Bit 0	$0019
Alternate Timer Register High (ATRH)	Bit 7	6	5	4	3	2	1	Bit 0	$001A
Alternate Timer Register Low (ATRL)	Bit 7	6	5	4	3	2	1	Bit 0	$001B

Figure 8-2. Timer I/O Register Summary

Latching values into the input capture registers at successive edges of the same polarity measures the period of the input signal on the TCAP pin. Latching the counter values at successive edges of opposite polarity measures the pulse width of the signal. Figure 8-3 shows the logic of the input capture function.

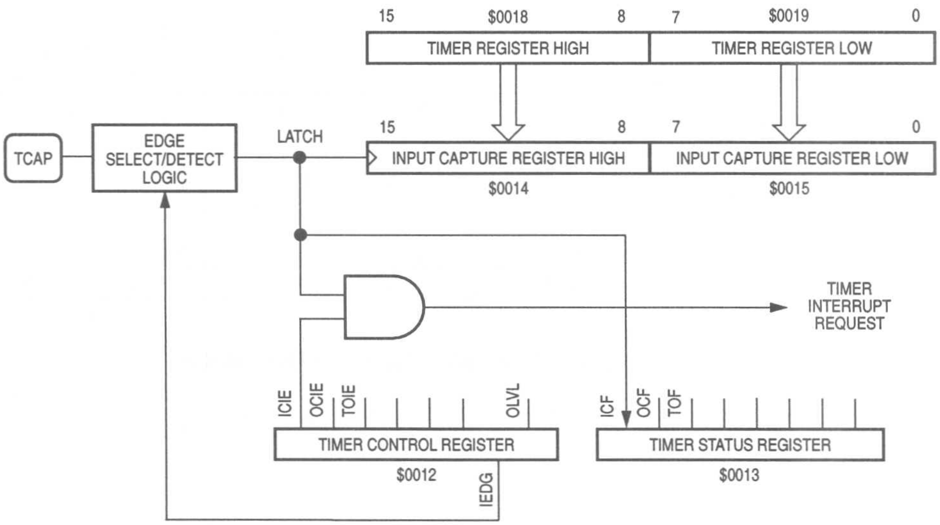


Figure 8-3. Input Capture Operation

8.2.2 Output Compare

The output compare function can generate an output signal when the 16-bit counter reaches a selected value. Software writes the selected value into the output compare registers. On every fourth internal clock cycle the output compare circuitry compares the value of the counter to the value written in the output compare registers. When a match occurs, the timer transfers the programmable output level bit (OLVL) from the timer control register to the output compare pin (TCMP).

Software can use the output compare register to measure time periods, to generate timing delays, or to generate a pulse of specific duration or a pulse train of specific frequency and duty cycle on the TCMP pin. Figure 8-4 shows the logic of the output compare function.

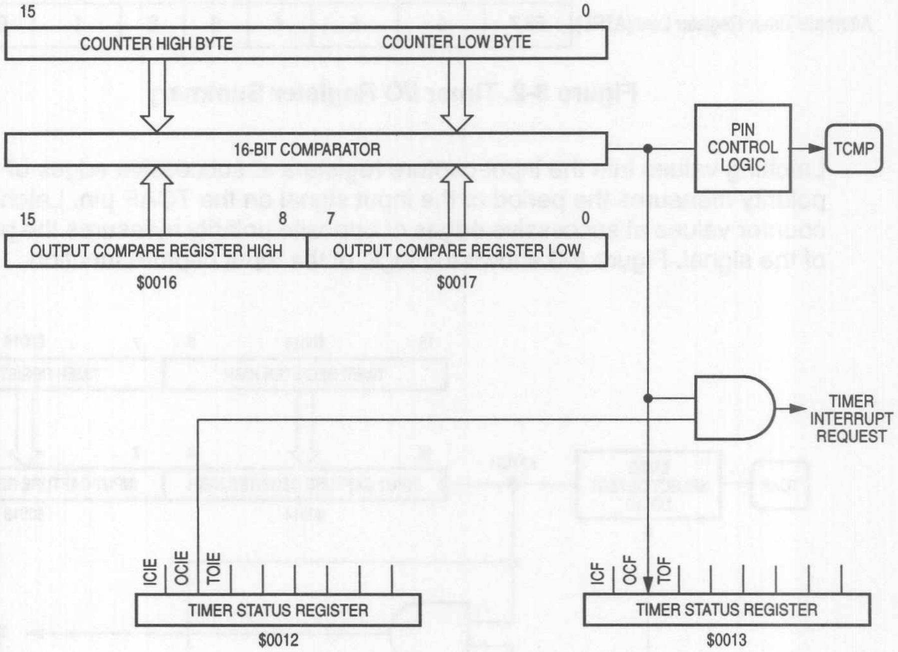


Figure 8-4. Output Compare Operation

8.3 Timer I/O Registers

The following registers control and monitor the timer operation:

- Timer control register (TCR)
- Timer status register (TSR)
- Timer registers (TRH and TRL)
- Alternate timer registers (ATRH and ATRL)
- Input capture registers (ICRH and ICRL)
- Output compare registers (OCRH and OCRL)

8.3.1 Timer Control Register (TCR)

The timer control register as shown in Figure 8-5 performs the following functions:

- Enables input capture interrupts
- Enables output compare interrupts
- Enables timer overflow interrupts
- Controls the active edge polarity of the TCAP signal
- Controls the active level of the TCMP output

		Bit 7	6	5	4	3	2	1	Bit 0
TCR $0012	Read: Write:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
	Reset:	0	0	0	0	0	0	U	0

U = Unaffected

Figure 8-5. Timer Control Register (TCR)

ICIE — Input Capture Interrupt Enable

This read/write bit enables interrupts caused by an active signal on the TCAP pin. Reset clears the ICIE bit.

1 = Input capture interrupts enabled
0 = Input capture interrupts disabled

OCIE — Output Compare Interrupt Enable

This read/write bit enables interrupts caused by an active signal on the TCMP pin. Reset clears the OCIE bit.

1 = Output compare interrupts enabled
0 = Output compare interrupts disabled

TOIE — Timer Overflow Interrupt Enable

This read/write bit enables interrupts caused by a timer overflow. Reset clears the TOIE bit.
1 = Timer overflow interrupts enabled
0 = Timer overflow interrupts disabled

IEDG — Input Edge

The state of this read/write bit determines whether a positive or negative transition on the TCAP pin triggers a transfer of the contents of the timer register to the input capture registers. Reset has no effect on the IEDG bit.
1 = Positive edge (low to high transition) triggers input capture
0 = Negative edge (high to low transition) triggers input capture

OLVL — Output Level Bit

The state of this read/write bit determines whether a logic one or a logic zero appears on the TCMP pin when a successful output compare occurs. Reset clears the OLVL bit.
1 = TCMP goes high on output compare
0 = TCMP goes low on output compare

Bits 4–2 — Not used; these bits always read zero.

8.3.2 Timer Status Register (TSR)

This read-only register shown in Figure 8-6 contains flags for the following events:

- An active signal on the TCAP pin, transferring the contents of the timer registers to the input capture registers
- A match between the 16-bit counter and the output compare registers, transferring the OLVL bit to the TCMP pin
- A timer rollover from $FFFF to $0000

		Bit 7	6	5	4	3	2	1	Bit 0
TSR $0013	Read:	ICF	OCF	TOF	0	0	0	0	0
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	U	U	U	0	0	0	0	0

▨ = Unimplemented U = Unaffected

Figure 8-6. Timer Status Register (TSR)

ICF — Input Capture Flag

The ICF bit is automatically set when an edge of the selected polarity occurs on the TCAP pin. Clear the ICF bit by reading the timer status register with ICF set and then reading the low byte ($0015) of the input capture registers. Reset has no effect on ICF.

1 = Input capture
0 = No input capture

OCF — Output Compare Flag

The OCF bit is set automatically when the value of the timer registers matches the contents of the output compare registers. Clear the OCF bit by reading the timer status register with OCF set, and then reading the low byte ($0017) of the output compare registers. Reset has no effect on OCF.

1 = Output compare
0 = No output compare

TOF — Timer Overflow Flag

The TOF bit is automatically set when the 16-bit counter rolls over from $FFFF to $0000. Clear the TOF bit by reading the timer status register with TOF set and then reading the low byte ($0019) of the timer registers. Reset has no effect on TOF.

1 = Timer overflow
0 = No timer overflow

Bits 4–0 — Not used; these bits always read zero.

8.3.3 Timer Registers (TRH and TRL)

The read-only timer registers shown in Figure 8-7 contain the current high and low bytes of the 16-bit counter. Reading TRH before reading TRL causes TRL to be latched until TRL is read. Reading TRL after reading the timer status register clears the timer overflow flag bit (TOF). Writing to the timer registers has no effect.

		Bit 7	6	5	4	3	2	1	Bit 0
TRH $0018	Read:	Bit 15	14	13	12	11	10	9	Bit 8
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	Reset initializes TRH to $FF							
		Bit 7	6	5	4	3	2	1	Bit 0
TRL $0019	Read	Bit 7	6	5	4	3	2	1	Bit 0
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	Reset initializes TRL to $FC							

▨ = Unimplemented

Figure 8-7. Timer Registers (TRH and TRL)

Reading TRH returns the current value of the high byte of the counter and causes the low byte to be latched into a buffer, as shown in Figure 8-8. The buffer value remains fixed even if the high byte is read more than once. Reading TRL reads the transparent low byte buffer and completes the read sequence of the timer registers.

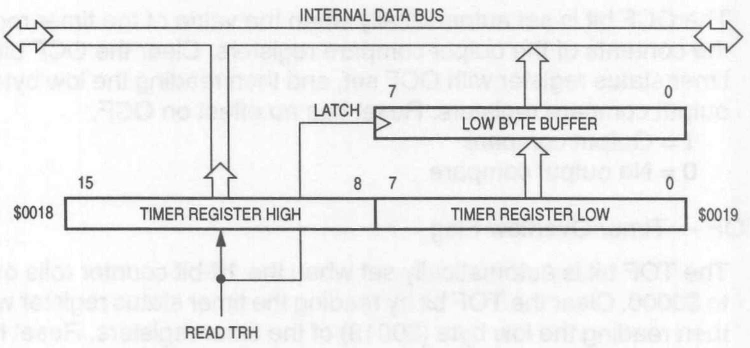


Figure 8-8. Timer Register Reads

NOTE

To prevent interrupts from occurring between readings of TRH and TRL, set the interrupt mask (I bit) in the condition code register before reading TRH, and clear the mask after reading TRL.

8.3.4 Alternate Timer Registers (ATRH and ATRL)

The alternate timer registers shown in Figure 8-9 contain the current high and low bytes of the 16-bit counter. Reading ATRH before reading ATRL causes ATRL to be latched until ATRL is read. Reading does not affect the timer overflow flag (TOF). Writing to the alternate timer registers has no effect.

		Bit 7	6	5	4	3	2	1	Bit 0
ATRH $001A	Read:	Bit 15	14	13	12	11	10	9	Bit 8
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	Reset initializes ATRH to $FF							
		Bit 7	6	5	4	3	2	1	Bit 0
ATRL $001B	Read:	Bit 7	6	5	4	3	2	1	Bit 0
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	Reset initializes ATRL to $FC							

▨ = Unimplemented

Figure 8-9. Alternate Timer Registers (ATRH and ATRL)

Reading ATRH returns the current value of the high byte of the counter and causes the low byte to be latched into a buffer, as shown in Figure 8-10.

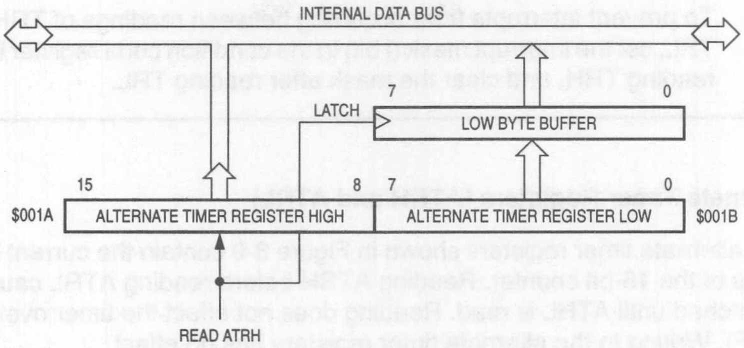


Figure 8-10. Alternate Timer Register Reads

NOTE

To prevent interrupts from occurring between readings of ATRH and ATRL, set the interrupt mask (I bit) in the condition code register before reading ATRH, and clear the mask after reading ATRL.

8.3.5 Input Capture Registers (ICRH and ICRL)

When a selected edge occurs on the TCAP pin, the current high and low bytes of the 16-bit counter are latched into the read-only input capture registers shown in Figure 8-11. Reading ICRH before reading ICRL inhibits further captures until ICRL is read. Reading ICRL after reading the timer status register clears the input capture flag (ICF). Writing to the input capture registers has no effect.

		Bit 7	6	5	4	3	2	1	Bit 0
ICRH $0014	Read:	Bit 15	14	13	12	11	10	9	Bit 8
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	Unaffected by reset							
		Bit 7	6	5	4	3	2	1	Bit 0
ICRL $0015	Read:	Bit 7	6	5	4	3	2	1	Bit 0
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	Unaffected by reset							

▨ = Unimplemented

Figure 8-11. Input Capture Registers (ICRH and ICRL)

NOTE

To prevent interrupts from occurring between readings of ICRH and ICRL, set the interrupt mask (I bit) in the condition code register before reading ICRH and clear the mask after reading ICRL.

8.3.6 Output Compare Registers (OCRH and OCRL)

When the value of the 16-bit counter matches the value in the read/write output compare registers shown in Figure 8-12, the planned TCMP pin action takes place. Writing to OCRH before writing to OCRL inhibits timer compares until OCRL is written. Reading or writing to OCRL after reading the timer status register clears the output compare flag (OCF).

		Bit 7	6	5	4	3	2	1	Bit 0
OCRH $0016	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8
	Reset:	Unaffected by reset							

		Bit 7	6	5	4	3	2	1	Bit 0
OCRL $0017	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
	Reset:	Unaffected by reset							

Figure 8-12. Output Compare Registers

To prevent OCF from being set between the time it is read and the time the output compare registers are updated, use the following procedure:

1. Disable interrupts by setting the I bit in the condition code register.
2. Write to OCRH. Compares are now inhibited until OCRL is written.
3. Clear bit OCF by reading the timer status register (TSR).
4. Enable the output compare function by writing to OCRL.
5. Enable interrupts by clearing the I bit in the condition code register.

SECTION 9
EPROM/OTPROM (PROM)

9.1 Introduction

This section describes erasable programmable read-only memory/one-time programmable read-only memory (EPROM/OTPROM (PROM)) programming.

9.2 EPROM/OTPROM (PROM) Programming

The internal PROM can be programmed efficiently using the Motorola MC68HC05PGMR-2 programmer board, which can be purchased from a Motorola-authorized distributor. The user can program the MCU using this printed circuit board (PCB) in conjunction with an EPROM device already programmed with user code.

Only standalone programming is discussed in this section. For more information concerning the MC68HC05PGMR and its usages, contact your local Motorola representative for a copy of MC68HC05PGMR2/D1, *MC68HC05PGMR Programmer Board User's Manual #2*.

Refer to Figure 9-1 for an EPROM programming flowchart and to Figure 9-2 for a schematic of the MC68HC05PGMR PCB.

To program the PROM MCU, the MCU is installed in the PCB, along with an EPROM device programmed with user code; the MCU is then subjected to a series of routines. The routines necessary to program, verify, and secure the PROM MCU are:

- Program and Verify PROM
- Verify PROM Contents (Only)
- Secure PROM and Verify
- Secure PROM and Dump (through SCI)

Other board routines available to the user are:

- Load Program into RAM and Execute
- Execute Program in RAM
- Dump PROM Contents (Binary Upload)

The user first configures the MCU for the bootstrap mode of operations by installing a fabricated jumper across pins 1 and 2 of the board's mode select header, J1. Next, the board's mode switches (S3, S4, S5, and S6) are set to determine the routine to be executed after the next reset, as shown in Table 9-1.

Table 9-1. PROM Programming Routines

Routine	S3	S4	S5	S6
Program and Verify PROM	Off	Off	Off	Off
Verify PROM Contents (Only)	Off	Off	On	Off
Secure PROM Contents and Verify	On	Off	On	Off
Secure PROM Contents and Dump	On	On	On	Off
Load Program into RAM and Execute	Off	On	Off	Off
Execute Program in RAM	Off	Off	Off	On
Dump PROM Contents	Off	On	On	Off

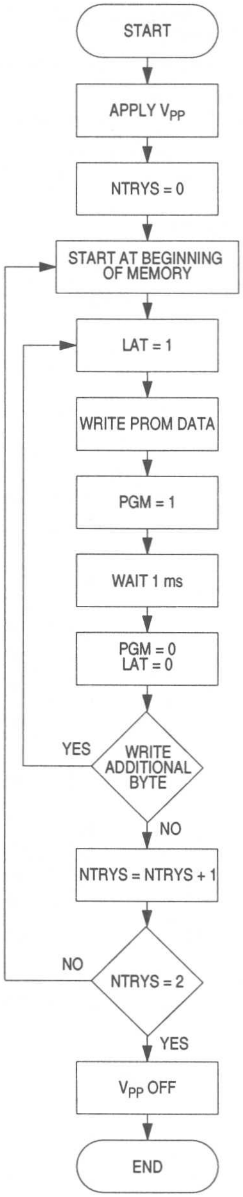
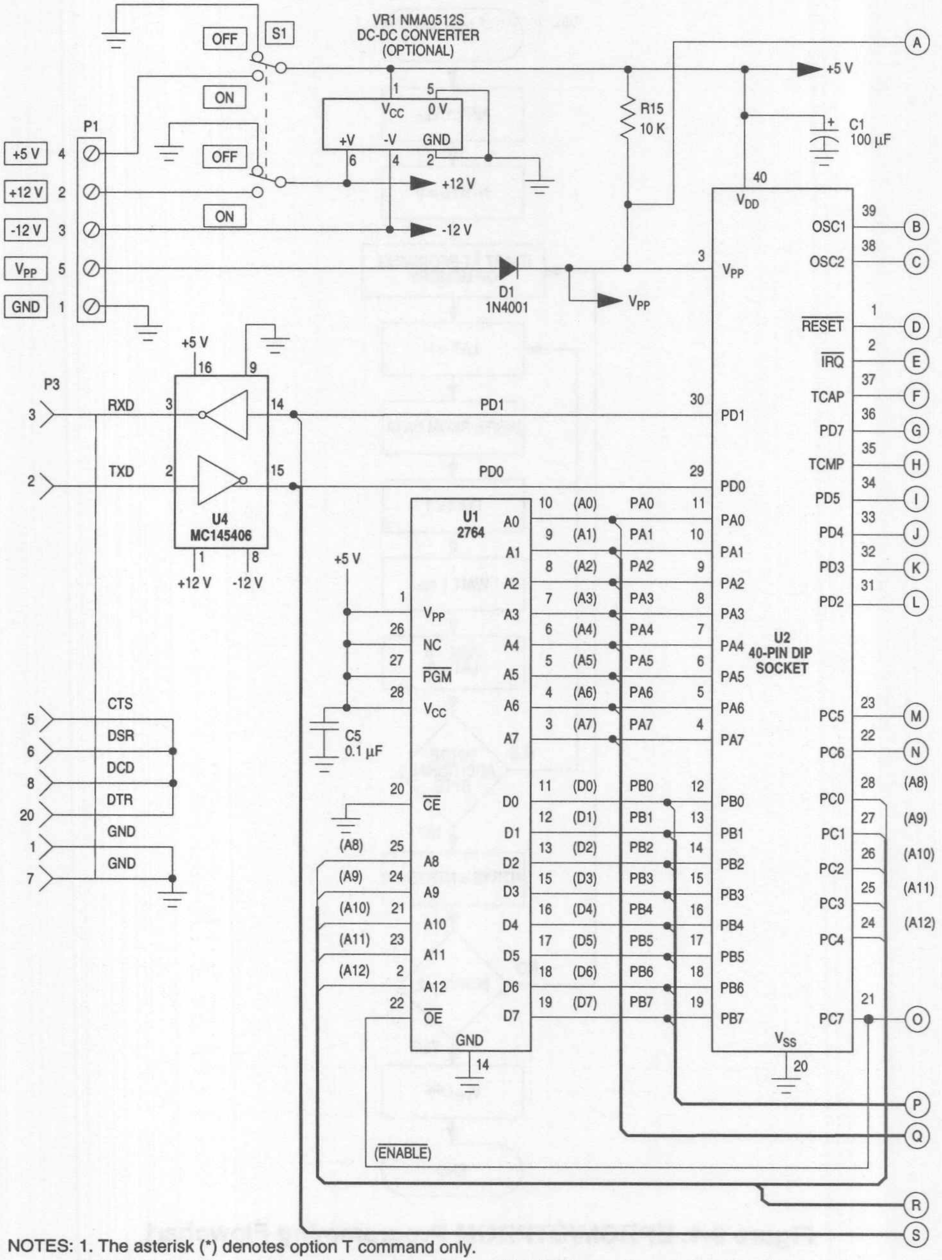
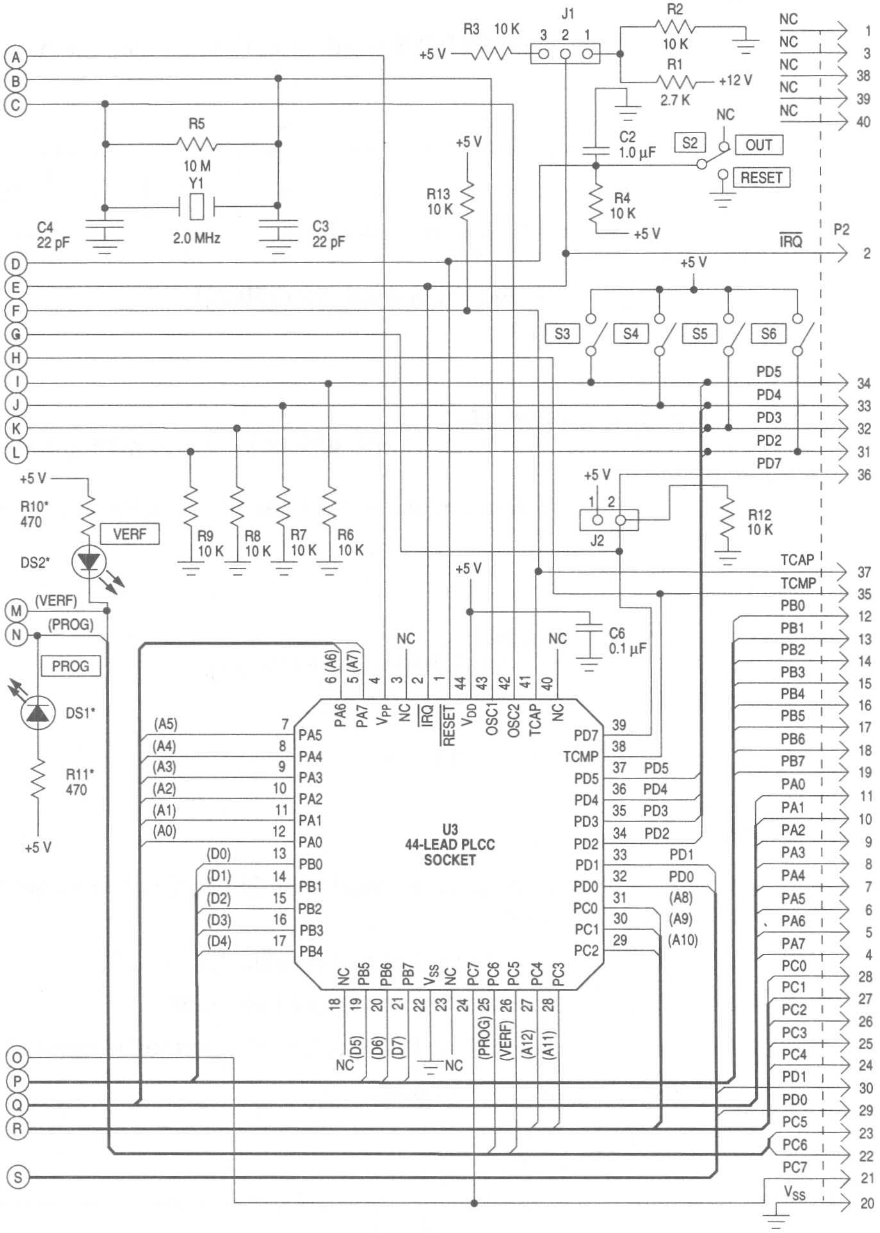


Figure 9-1. EPROM/OTPROM Programming Flowchart



NOTES: 1. The asterisk (*) denotes option T command only.
2. Unless otherwise specified, resistors are in ohms, ±5% 1/4 W; capacitors are in µF; voltages are dc.
3. Device type numbers shown in circuit are for reference only. Device type number varies with manufacturer.

Figure 9-2. PROM Programming Circuit



Ref. Description	Device Type	Notes	Ground	+ 5 V	+ 12 V	– 12 V	V_{PP}
U1	2764	8 K x 8-Bit EPROM	14, 20	1, 26, 27, 28			
U2	MCU	40-Pin DIP Socket	20	40			3
U3	MCU	44-Lead PLCC Socket	22	44			4
U4	MC145406	Driver/Receiver	9	16	1	8	
VR1	NMA0512S	DC-DC Converter	2.5	1	6	4	

9.2.1 Program Register (PROG)

The program register shown in Figure 9-3 is used for PROM programming.

		Bit 7	6	5	4	3	2	1	Bit 0
PROG $001C	Read: Write:	0	0	0	0	0	LAT	0	PGM
	Reset:	0	0	0	0	0	0	0	0

Figure 9-3. Program Register (PROG)

LAT — Latch Enable

This bit is both readable and writable.

1 = Enables PROM data and address bus latches for programming on the next byte write cycle.

0 = Latch disabled. PROM data and address buses are unlatched for normal CPU operations.

PGM — Program

If LAT is cleared, PGM cannot be set.

1 = Enables V_{PP} power to the PROM for programming.

0 = V_{PP} is disabled.

Bits 1 and 3–7 — Not used; always read zero.

9.2.2 Preprogramming Steps

Before programming the PROM using an MC68HC05PGMR PCB in standalone mode, the user should ensure that:

- A jumper is installed on pins 1 and 2 of mode select header J1.
- An EPROM is programmed with the necessary user code.
- The erasure window (if any) of the device to be programmed is covered.
- V_{DD} of +5 Vdc is available on the board.
- V_{PP} is available on the board.

CAUTION

If the V_{PP} level at the MCU exceeds +16 Vdc, then the MC68HC705C4A MCU device will suffer permanent damage.

Once the above conditions are met, the user should take the following steps before beginning programming:

1. Remove the V_{PP} power source.
2. Set switch 1 in the OFF position (removes V_{DD}).
3. Place the programmed EPROM in socket U1.
4. Insert the erased PROM MCU device to be programmed in the proper socket:
   - MC68HC705C8S or MC68HC705C8P in socket U2 (40-pin DIP), or,
   - MC68HC705C8FN in socket U3 (44-pin PLCC) with the device notch at the upper right corner of the socket.
5. Set switch S2 in the RESET position.

NOTE

No PROM MCU should be inserted in or removed from its board socket (U2 or U3) while V_{PP} (P1, slot 5) or V_{DD} (switch 1) is active on the board.

9.3 PROM Programming Routines

Following are the routines necessary to program, verify, and secure the PROM device, and other routines available to the user.

9.3.1 Program and Verify PROM

The program and verify PROM routine copies the contents of the external EPROM into the MCU PROM with direct correspondence between the addresses. Memory addresses in the MCU that are not implemented in PROM are skipped. Unprogrammed addresses in the EPROM being copied should contain $00 bytes to speed up the programming process.

To run the program and verify PROM routine on the PROM MCU, take the following steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Restore the V_{PP} power source.
3. Set switches S3, S4, S5, and S6 in the OFF position (selects proper routine).
4. Set switch 2 in the OUT position (routine is activated).
   The red light-emitting diode (LED) is illuminated, showing that the programming part of the routine is running. The LED goes out when programming is finished. The verification part of the routine now begins. When the green LED is illuminated, verification is successfully completed and the routine is finished.
5. Set switch 2 in the RESET position.

At this point, if no other MCU is to be programmed or secured, remove V_{PP} power from the board. If another routine is to be performed on the MCU being programmed, the user can then set switches S3, S4, S5, and S6 to the positions necessary to select the next routine, and begin the routine by setting switch 2 to the OUT position. If no other routine is to be performed, remove V_{DD} from the board and remove the MCU from the programming socket.

9.3.2 Verify PROM Contents

The verify PROM contents routine is normally run automatically after the PROM is programmed. Direct entry to this routine causes the PROM contents of the MCU to be compared to the contents of the external memory locations of the EPROM at the same addresses.

To invoke the verify PROM contents routine of the MCU, take the following steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.

3. Set switches S3, S4, and S6 in the OFF position.
4. Set S5 in the ON position.
5. Set switch 2 in the OUT position (routine is activated).
   The red LED is not illuminated during this routine, since no programming takes place. If verification fails, the routine halts with the failing address in the external memory bus. When the green LED is illuminated, verification is completed successfully and the routine is finished.
6. Set switch 2 in the RESET position.

At this point, if another routine is to be performed on the MCU being programmed, the user can set switches S3, S4, S5, and S6 to the positions necessary to select the next routine, and move switch S2 to the OUT position to start routine. If no other routine is to be performed, remove V_{DD} from the board and remove the MCU from the programming socket.

9.3.3 Secure PROM

The secure PROM routines are used after the PROM is successfully programmed and verified. Only the SEC bit of the OPTION register ($1FDF) is programmed, but V_{PP} is necessary. Once this bit is programmed, PROM is secure and can be neither verified nor dumped.

9.3.4 Secure PROM and Verify

This routine is used after the PROM is programmed successfully to verify the contents of the MCU PROM against the contents of the EPROM and then to secure the PROM. To accomplish this routine, take the following steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Restore V_{PP} power to the programming board.
3. Set switches S4 and S6 in the OFF position.
4. Set switches S3 and S5 in the ON position.
5. Set switch 2 in the OUT position (routine is activated).
   Execution time for this routine is about one second.
6. Set switch 2 in the RESET position when the routine is completed.

No LED is illuminated during this routine. Note also that the end of the routine does not mean that the SEC bit was verified. To ensure that security is properly enabled, attempt to perform another verify routine. If the green LED does not light, the PROM has been secured properly.

9.3.5 Secure PROM and Dump

This routine is used after the PROM is successfully programmed to dump the contents of the MCU PROM through the SCI (binary upload) and then to secure the PROM. To accomplish this routine, take the following steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Restore V_{PP} power to the programming board.
3. Set switch S6 in the OFF position.
4. Set switches S3, S4, and S5 in the ON position.
5. Set switch 2 in the OUT position (routine is activated). Execution time for this routine is about one second.
6. Set switch 2 in the RESET position when the routine is completed.

No LED is illuminated during this routine. Further, the end of the routine does not mean that the SEC bit was verified. To ensure that security is properly enabled, attempt to perform another verify routine. If the green LED does not light, the PROM has been secured properly.

9.3.6 Load Program into RAM and Execute

In the load program in RAM and execute routine, user programs are loaded via the SCI port and then executed. Data is loaded sequentially starting at address $0050. After the last byte is loaded, control is transferred to the RAM program starting at $0051. The first byte loaded is the count of the total number of bytes in the program plus the count byte. The program starts at location $0051 in RAM. During initialization, the SCI is configured for eight data bits and one stop bit. The baud rate is 4800 with a 2-MHz crystal or 9600 with a 4-MHz crystal.

To load a program into RAM and execute it, take the following steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switches S3, S5, and S6 in the OFF position.
4. Set switch S4 in the ON position.
5. Set switch 2 in the OUT position (routine is activated).

The downloaded program starts executing as soon as the last byte is received by the SCI.

Execution of the routine can be held off by setting the byte count in the count byte (the first byte loaded) to a value greater than the number of bytes to be loaded. After loading the last byte, the firmware waits for more data. Program execution does not begin. At this point, placing switch 2 in the RESET position resets the MCU with the RAM data intact. Any other routine can be entered, including the one to execute the program in RAM, simply by setting switches S3–S6 as necessary to select the desired routine, then setting switch 2 in the OUT position.

9.3.7 Execute Program in RAM

This routine allows the MCU to transfer control to a program previously loaded in RAM. This program is executed once bootstrap mode is entered, if switch S6 is in the ON position and switch 2 is in the OUT position, without any firmware initialization. The program must start at location $0051 to be compatible with the load program in RAM routine.

To run the execute program in RAM routine, take the following steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switch S6 in the OFF position.
4. Switches S3, S4, and S5 can be in either position.
5. Set switch 2 in the OUT position (routine is activated).

9.3.8 Dump PROM Contents

In the dump PROM contents routine, the PROM contents are dumped sequentially to the SCI output, provided the PROM has not been secured. The first location sent is $0020 and the last location sent is $1FFF. Unused locations are skipped so that no gaps exist in the data stream. The external memory address lines indicate the current location being sent. Data is sent with eight data bits and one stop bit at 4800 baud with a 2-MHz crystal or 9600 baud with a 4-MHz crystal.

To run the dump PROM contents routine, take the following steps:

1. Set switch 1 in the ON position (restores V_{DD}).
2. Connect V_{PP} to V_{DD}.
3. Set switches S3 and S6 in the OFF position.
4. Set switches S4 and S5 in the ON position.
5. Set switch 2 in the OUT position (routine is activated).
6. Once PROM dumping is complete, set switch 2 in the RESET position.

9.4 Control Registers

The following paragraphs describe three registers that control memory configuration, PROM security, and IRQ edge or level sensitivity; port B pullups; and non-programmable COP enable/disable.

9.4.1 Option Register (Option)

The option register shown in Figure 9-4 is used to select the IRQ sensitivity, enable the PROM security, and select the memory configuration.

		Bit 7	6	5	4	3	2	1	Bit 0
Option $1FDF	Read: Write:	0	0	0	0	SEC*		IRQ	0
	Reset:	0	0	0	0	*	U	1	0

▨ = Unimplemented   U = Unaffected

*Implemented as an EPROM cell

Figure 9-4. Option Register (Option)

SEC — Security

This bit is implemented as an EPROM cell and is not affected by reset.

1 = Security enabled.

0 = Security off; bootloader able to be enabled.

IRQ — Interrupt Request Pin Sensitivity

IRQ is set only by reset, but can be cleared by software. This bit can only be written once.

1 = $\overline{\text{IRQ}}$ pin is both negative edge- and level-sensitive.

0 = $\overline{\text{IRQ}}$ pin is negative edge-sensitive only.

Bits 7–4 and 0 — Not used; always read zero.

Bit 2 — Unaffected by reset; reads either one or zero.

9.4.2 Mask Option Register 1 (MOR1)

MOR1 shown in Figure 9-5 is an EPROM register that enables the port B pullup devices. Data from MOR1 is latched on the rising edge of the voltage on the $\overline{\text{RESET}}$ pin. (See **4.2.3 Port B Interrupts**.)

		Bit 7	6	5	4	3	2	1	Bit 0
MOR1 $1FF0	Read: Write:	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0/ COPC
	Reset:	Unaffected by reset							
	Erased:	0	0	0	0	0	0	0	0

Figure 9-5. Mask Option Register 1 (MOR1)

PBPU7–PBPU0 — Port B Pullup Enable Bits 7–0

These EPROM bits enable the port B pullup devices.
1 = Port B pullups enabled
0 = Port B pullups disabled

NOTE

PBPU0/COPC programmed to a one enables the port B pullup bit. This bit is also used to clear the COP. Writing to this bit to clear the COP will not affect the state of the port B pull-up (bit 0). (See **5.2.3 COP Watchdog Reset**.)

9.4.3 Mask Option Register 2 (MOR2)

MOR2 shown in Figure 9-6 is an EPROM register that enables the non-programmable COP watchdog. Data from MOR2 is latched on the rising edge of the voltage on the $\overline{\text{RESET}}$ pin. (See **5.2.3 COP Watchdog Reset**.)

		Bit 7	6	5	4	3	2	1	Bit 0
MOR2 $1FF1	Read:	▨	▨	▨	▨	▨	▨	▨	NCOPE
	Write:	▨	▨	▨	▨	▨	▨	▨	
	Reset:	Unaffected by reset							
	Erased:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 9-6. Mask Option Register 2 (MOR2)

NCOPE— Non-Programmable COP Watchdog Enable

This EPROM bit enables the non-programmable COP watchdog.

1 = Non-programmable COP watchdog enabled

0 = Non-programmable COP watchdog disabled

9.5 EPROM Erasing

The erased state of an EPROM or OTPROM byte is $00. EPROM devices can be erased by exposure to a high intensity ultraviolet (UV) light with a wavelength of 2537 Å. The recommended erasure dosage (UV intensity on a given surface area times exposure time) is 15 Ws/cm^2. UV lamps should be used without short-wave filters, and the EPROM device should be positioned about one inch from the UV source.

OTPROM devices are shipped in an erased state. Once programmed, they cannot be erased. Electrical erasing procedures cannot be performed on either EPROM or OTPROM devices.

SECTION 10
SERIAL COMMUNICATIONS INTERFACE (SCI)

10.1 Introduction

The SCI module allows high-speed asynchronous communication with peripheral devices and other MCUs.

10.2 Features

Features of the SCI module include:

- Standard Mark/Space Non-Return-to-Zero Format
- Full Duplex Operation
- 32 Programmable Baud Rates
- Programmable 8-Bit or 9-Bit Character Length
- Separately Enabled Transmitter and Receiver
- Two Receiver Wakeup Methods:
  - Idle Line Wakeup
  - Address Mark Wakeup
- Interrupt-Driven Operation Capability with Five Interrupt Flags:
  - Transmitter Data Register Empty
  - Transmission Complete
  - Receiver Data Register Full
  - Receiver Overrun
  - Idle Receiver Input
- Receiver Framing Error Detection
- 1/16 Bit-Time Noise Detection

10.3 SCI Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in Figure 10-1.

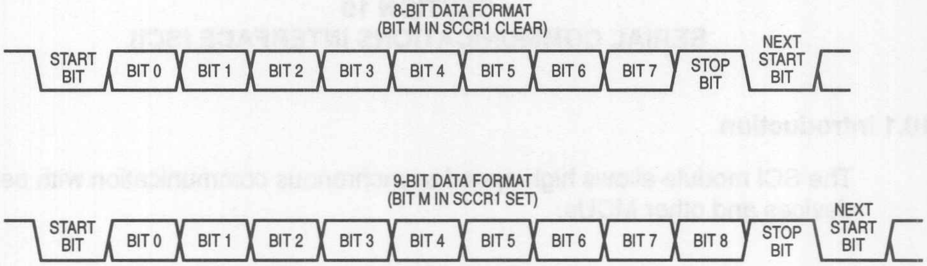


Figure 10-1. SCI Data Format

10.4 SCI Operation

The SCI allows full-duplex, asynchronous, RS232 or RS422 serial communication between the MCU and remote devices, including other MCUs. The transmitter and receiver of the SCI operate independently, although they use the same baud-rate generator. The following paragraphs describe the operation of the SCI transmitter and receiver.

10.4.1 Transmitter

Figure 10-2 shows the structure of the SCI transmitter. Figure 10-3 is a summary of the SCI transmitter I/O registers.

- **Character Length** — The transmitter can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCCR1) determines character length. When transmitting 9-bit data, bit T8 in SCCR1 is the ninth bit (bit 8).
- **Character Transmission** — During transmission, the transmit shift register shifts a character out to the PD1/TDO pin. The SCI data register (SCDR) is the write-only buffer between the internal data bus and the transmit shift register.

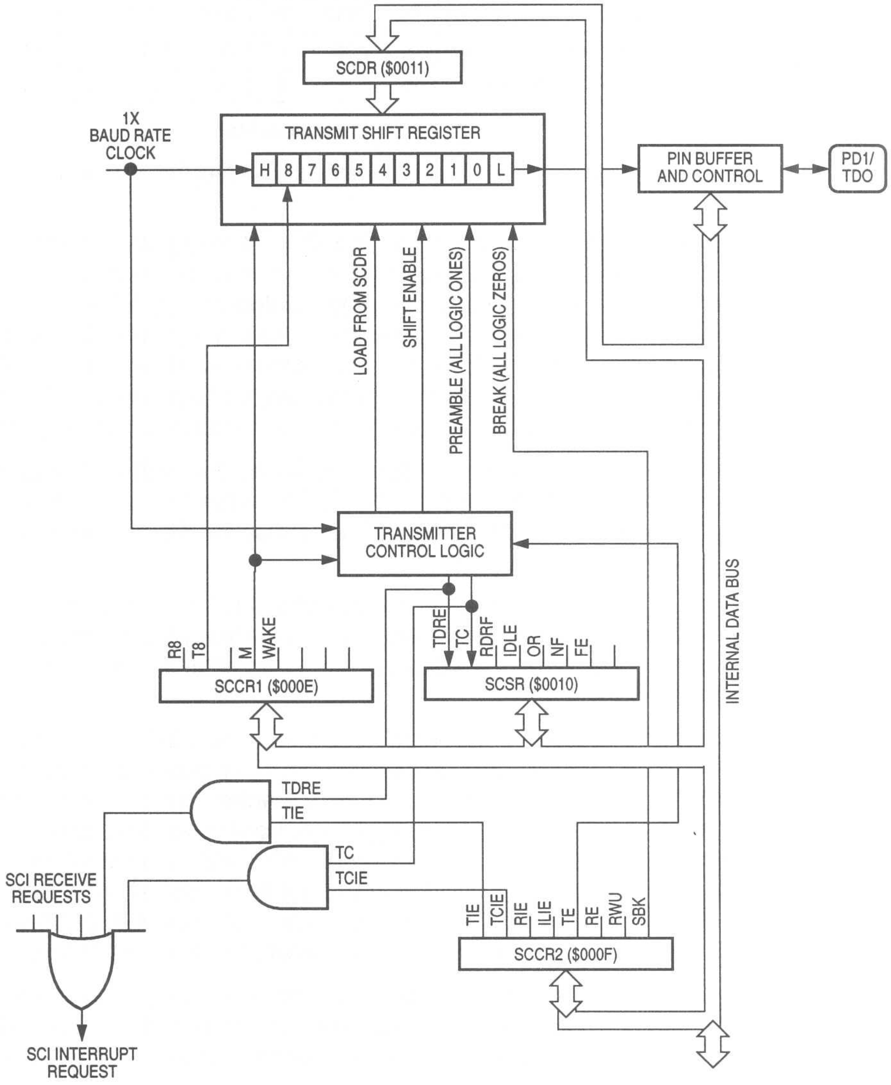


Figure 10-2. SCI Transmitter

Register Name	Bit 7	6	5	4	3	2	1	Bit 0	Addr.
Baud Rate Register (Baud)	▨	▨	SCP1	SCP0	▨	SCR2	SCR1	SCR0	$000D
SCI Control Register 1 (SCCR1)	R8	T8	▨	M	WAKE	▨	▨	▨	$000E
SCI Control Register 2 (SCCR2)	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	$000F
SCI Status Register (SCSR)	TDRE	TC	RDRF	IDLE	OR	NF	FE	▨	$0010
SCI Data Register (SCDR)	Bit 7	6	5	4	3	2	1	Bit 0	$0011

▨ = Unimplemented

Figure 10-3. SCI Transmitter I/O Register Summary

Writing a logic one to the TE bit in SCI control register 2 (SCCR2) and then writing data to the SCDR begins the transmission. At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of logic ones. After the preamble shifts out, the control logic transfers the SCDR data into the shift register. A logic zero start bit automatically goes into the least significant bit position of the shift register, and a logic one stop bit goes into the most significant bit position.

When the data in the SCDR transfers to the transmit shift register, the transmit data register empty (TDRE) flag in the SCI status register (SCSR) becomes set. The TDRE flag indicates that the SCDR can accept new data from the internal data bus.

When the shift register is not transmitting a character, the PD1/TDO pin goes to the idle condition, logic one. If software clears the TE bit during the idle condition, and while TDRE is set, the transmitter relinquishes control of the PD1/TDO pin.

- **Break Characters** — Writing a logic one to the SBK bit in SCCR2 loads the shift register with a break character. A break character contains all logic zeros and has no start and stop bits. Break character length depends on the M bit in SCCR1. As long as SBK is at logic one, transmitter logic continuously loads break characters into the shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic one. The automatic logic one at the end of a break character is to guarantee the recognition of the start bit of the next character.
- **Idle Characters** — An idle character contains all logic ones and has no start or stop bits. Idle character length depends on the M bit in SCCR1. The preamble is a synchronizing idle character that begins every transmission.

Clearing the TE bit during a transmission relinquishes the PD1/TDO pin after the last character to be transmitted is shifted out. The last character may already be in the shift register, or waiting in the SCDR, or it may be a break character generated by writing to the SBK bit. Toggling TE from logic zero to logic one while the last character is in transmission generates an idle character (a preamble) that allows the receiver to maintain control of the PD1/TDO pin.

- **Transmitter Interrupts** — The following sources can generate SCI transmitter interrupt requests:
  - Transmit Data Register Empty (TDRE) — The TDRE bit in the SCSR indicates that the SCDR has transferred a character to the transmit shift register. TDRE is a source of SCI interrupt requests. The transmission complete interrupt enable bit (TCIE) in SCCR2 is the local mask for TDRE interrupts.
  - Transmission Complete (TC) — The TC bit in the SCSR indicates that both the transmit shift register and the SCDR are empty and that no break or idle character has been generated. TC is a source of SCI interrupt requests. The transmission complete interrupt enable bit (TCIE) in SCCR2 is the local mask for TC interrupts.

10.4.2 Receiver

Figure 10-4 shows the structure of the SCI receiver. Figure 10-5 is a summary of the SCI receiver I/O registers.

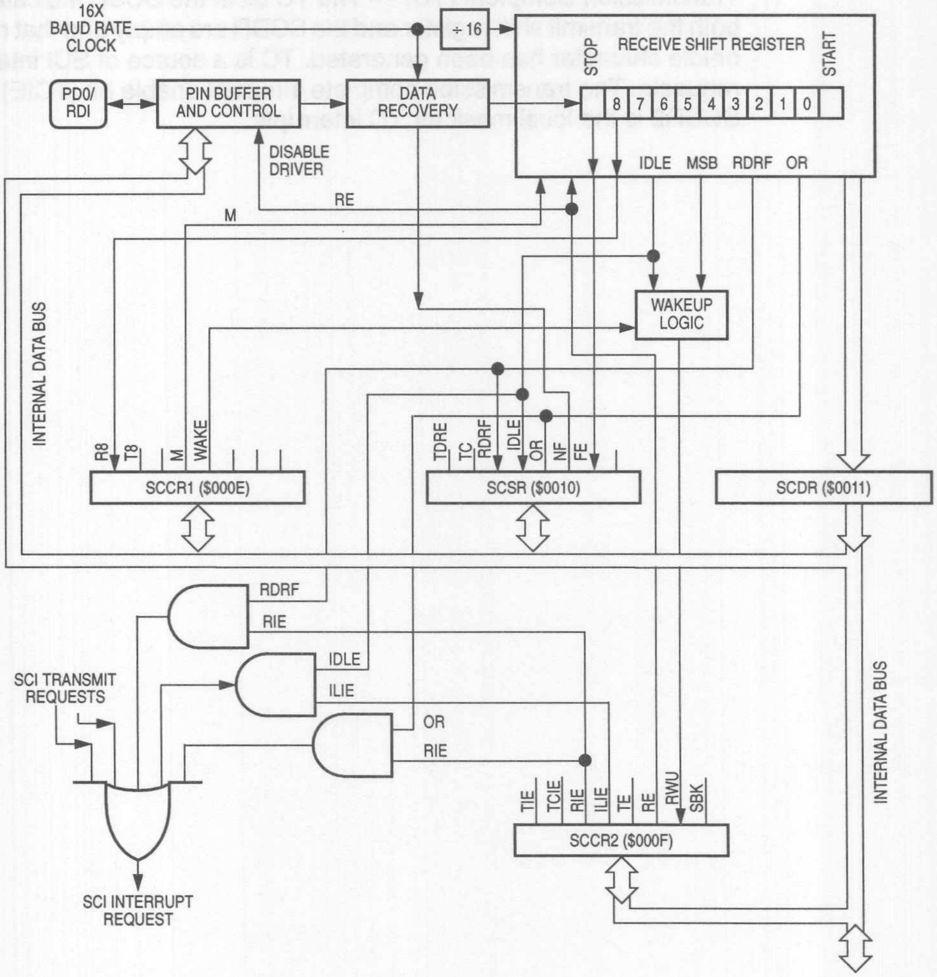


Figure 10-4. SCI Receiver

Register Name	Bit 7	6	5	4	3	2	1	Bit 0	Addr.
Baud Rate Register (Baud)	▨	▨	SCP1	SCP0	▨	SCR2	SCR1	SCR0	$000D
SCI Control Register 1 (SCCR1)	R8	T8	▨	M	WAKE	▨	▨	▨	$000E
SCI Control Register 2 (SCCR2)	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	$000F
SCI Status Register (SCSR)	TDRE	TC	RDRF	IDLE	OR	NF	FE	▨	$0010
SCI Data Register (SCDR)	Bit 7	6	5	4	3	2	1	Bit 0	$0011

▨ = Unimplemented

Figure 10-5. SCI Receiver I/O Register Summary

- **Character Length** — The receiver can accommodate either 8-bit or 9-bit data. The state of the M bit in SCI control register 1 (SCCR1) determines character length. When receiving 9-bit data, bit R8 in SCCR1 is the ninth bit (bit 8).
- **Character Reception** — During reception, the receive shift register shifts characters in from the PD0/RDI pin. The SCI data register (SCDR) is the read-only buffer between the internal data bus and the receive shift register.

  After a complete character shifts into the receive shift register, the data portion of the character is transferred to the SCDR, setting the receive data register full (RDRF) flag. The RDRF flag can be used to generate an interrupt.

- **Receiver Wakeup** — So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the MCU can be put into a standby state. Setting the receiver wakeup enable (RWU) bit in SCI control register 2 (SCCR2) puts the MCU into a standby state during which receiver interrupts are disabled.
  - Either of two conditions on the PD0/RDI pin can bring the MCU out of the standby state:
    Idle input line condition — If the PD0/RDI pin is at logic one long enough for 10 or 11 logic ones to shift into the receive shift register, receiver interrupts are again enabled.
  - Address mark — If a logic one occurs in the most significant bit position of a received character, receiver interrupts are again enabled.

  The state of the WAKE bit in SCCR1 determines which of the two conditions wakes up the MCU.
- **Receiver Noise Immunity** — The data recovery logic samples each bit 16 times to identify and verify the start bit and to detect noise. Any conflict between noise detection samples sets the noise flag (NF) in the SCSR. The NF bit is set at the same time that the RDRF bit is set.

- **Framing Errors** — If the data recovery logic does not detect a logic one where the stop bit should be in an incoming character, it sets the framing error (FE) bit in the SCSR. The FE bit is set at the same time that the RDRF bit is set.
- **Receiver Interrupts** — The following sources can generate SCI receiver interrupt requests:
  - Receive Data Register Full (RDRF) — The RDRF bit in the SCSR indicates that the receive shift register has transferred a character to the SCDR.
  - Receiver Overrun (OR) — The OR bit in the SCSR indicates that the receive shift register shifted in a new character before the previous character was read from the SCDR.
  - Idle Input (IDLE) — The IDLE bit in the SCSR indicates that 10 or 11 consecutive logic ones shifted in from the PD0/RDI pin.

10.5 SCI I/O Registers

The following I/O registers control and monitor SCI operation:

- SCI Data Register (SCDR)
- SCI Control Register 1 (SCCR1)
- SCI Control Register 2 (SCCR2)
- SCI Status Register (SCSR)

10.5.1 SCI Data Register (SCDR)

The SCI data register shown in Figure 10-6 is the buffer for characters received and for characters transmitted.

		Bit 7	6	5	4	3	2	1	Bit 0
SCDR $0011	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
	Reset:	Unaffected by reset							

Figure 10-6. SCI Data Register (SCDR)

10.5.2 SCI Control Register 1 (SCCR1)

SCI control register 1 shown in Figure 10-7 has the following functions:

- Stores ninth SCI data bit received and ninth SCI data bit transmitted
- Controls SCI character length
- Controls SCI wakeup method

SCCR1 $000E		Bit 7	6	5	4	3	2	1	Bit 0
	Read: Write:	R8	T8	▨	M	WAKE	▨	▨	▨
	Reset:	U	U		U	U			

▨ = Unimplemented U = Unaffected

Figure 10-7. SCI Control Register 1 (SCCR1)

R8 — Bit 8 (Received)

When the SCI is receiving 9-bit characters, R8 is the ninth bit of the received character. R8 receives the ninth bit at the same time that the SCDR receives the other eight bits. Reset has no effect on the R8 bit.

T8 — Bit 8 (Transmitted)

When the SCI is transmitting 9-bit characters, T8 is the ninth bit of the transmitted character. T8 is loaded into the transmit shift register at the same time that SCDR is loaded into the transmit shift register. Reset has no effect on the T8 bit.

M — Character Length

This read/write bit determines whether SCI characters are eight or nine bits long. The ninth bit can be used as an extra stop bit, as a receiver wakeup signal, or as a mark or space parity bit. Reset has no effect on the M bit.

1 = 9-bit SCI characters
0 = 8-bit SCI characters

WAKE —Wakeup Bit

This read/write bit determines which condition wakes up the SCI: a logic one (address mark) in the most significant bit position of a received character or an idle condition of the PD0/RDI pin. Reset has no effect on the WAKE bit.

1 = Address mark wakeup
0 = Idle line wakeup

10.5.3 SCI Control Register 2 (SCCR2)

SCI control register 2 shown in Figure 10-8 has the following functions:

- Enables the SCI receiver and SCI receiver interrupts
- Enables the SCI transmitter and SCI transmitter interrupts
- Enables SCI receiver idle interrupts
- Enables SCI transmission complete interrupts
- Enables SCI wakeup
- Transmits SCI break characters

		Bit 7	6	5	4	3	2	1	Bit 0
SCCR2 $000F	Read: Write:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	Reset:	0	0	0	0	0	0	0	0

Figure 10-8. SCI Control Register 2 (SCCR2)

TIE — Transmit Interrupt Enable

This read/write bit enables SCI interrupt requests when the TDRE bit becomes set. Reset clears the TIE bit.

1 = TDRE interrupt requests enabled
0 = TDRE interrupt requests disabled

TCIE — Transmission Complete Interrupt Enable

This read/write bit enables SCI interrupt requests when the TC bit becomes set. Reset clears the TCIE bit.

1 = TC interrupt requests enabled
0 = TC interrupt requests disabled

RIE — Receive Interrupt Enable

This read/write bit enables SCI interrupt requests when the RDRF bit or the OR bit becomes set. Reset clears the RIE bit.

1 = RDRF interrupt requests enabled
0 = RDRF interrupt requests disabled

ILIE — Idle Line Interrupt Enable

This read/write bit enables SCI interrupt requests when the IDLE bit becomes set. Reset clears the ILIE bit.
1 = IDLE interrupt requests enabled
0 = IDLE interrupt requests disabled

TE — Transmit Enable

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 logic ones from the transmit shift register to the PD1/TDO pin. Reset clears the TE bit.
1 = Transmission enabled
0 = Transmission disabled

RE — Receive Enable

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver and receiver interrupts but does not affect the receiver interrupt flags. Reset clears the RE bit.
1 = Receiver enabled
0 = Receiver disabled

RWU — Receiver Wakeup Enable

This read/write bit puts the receiver in a standby state. Typically, data transmitted to the receiver clears the RWU bit and returns the receiver to normal operation. The WAKE bit in SCCR1 determines whether an idle input or an address mark brings the receiver out of the standby state. Reset clears the RWU bit.
1 = Standby state
0 = Normal operation

SBK — Send Break

Setting this read/write bit continuously transmits break codes in the form of 10-bit or 11-bit groups of logic zeros. Clearing the SBK bit stops the break codes and transmits a logic one as a start bit. Reset clears the SBK bit.
1 = Break codes being transmitted
0 = No break codes being transmitted

10

10.5.4 SCI Status Register (SCSR)

The SCI status register shown in Figure 10-9 contains flags to signal the following conditions:

- Transfer of SCDR data to transmit shift register complete
- Transmission complete
- Transfer of receive shift register data to SCDR complete
- Receiver input idle
- Receiver overrun
- Noisy data
- Framing error

		Bit 7	6	5	4	3	2	1	Bit 0
SCSR $0010	Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	▨
	Write:	▨	▨	▨	▨	▨	▨	▨	
	Reset:	1	1	0	0	0	0	0	U

▨ = Unimplemented U = Unaffected

Figure 10-9. SCI Status Register (SCSR)

TDRE — Transmit Data Register Empty

This clearable, read-only bit is set when the data in the SCDR transfers to the transmit shift register. TDRE generates an interrupt request if the TIE bit in SCCR2 is also set. Clear the TDRE bit by reading the SCSR with TDRE set, and then writing to the SCDR. Reset sets the TDRE bit. Software must initialize the TDRE bit to logic zero to avoid an instant interrupt request when turning on the transmitter.

1 = SCDR data transferred to transmit shift register
0 = SCDR data not transferred to transmit shift register

TC — Transmission Complete

This clearable, read-only bit is set when the TDRE bit is set and no data, preamble, or break character is being transmitted. TC generates an interrupt request if the TCIE bit in SCCR2 is also set. Clear the TC bit by reading the SCSR with TC set and then writing to the SCDR. Reset sets the TC bit. Software must initialize the TC bit to logic zero to avoid an instant interrupt request when turning on the transmitter.

1 = No transmission in progress
0 = Transmission in progress

RDRF — Receive Data Register Full

This clearable, read-only bit is set when the data in the receive shift register transfers to the SCI data register. RDRF generates an interrupt request if the RIE bit in SCCR2 is also set. Clear the RDRF bit by reading the SCSR with RDRF set and then reading the SCDR. Reset clears the RDRF bit.

1 = Received data available in SCDR
0 = Received data not available in SCDR

IDLE — Receiver Idle

This clearable, read-only bit is set when 10 or 11 consecutive logic ones appear on the receiver input. IDLE generates an interrupt request if the ILIE bit in SCCR2 is also set. Clear the IDLE bit by reading the SCSR with IDLE set, and then reading the SCDR. Reset clears the IDLE bit.

1 = Receiver input idle
0 = Receiver input not idle

OR — Receiver Overrun

This clearable, read-only bit is set if the SCDR is not read before the receive shift register receives the next word. OR generates an interrupt request if the RIE bit in SCCR2 is also set. The data in the shift register is lost, but the data already in the SCDR is not affected. Clear the OR bit by reading the SCSR with OR set, and then reading the SCDR. Reset clears the OR bit.

1 = Receiver shift register full and RDRF = 1
0 = No receiver overrun

NF — Receiver Noise Flag

This clearable, read-only bit is set when noise is detected in data received in the SCI data register. Clear the NF bit by reading the SCSR, and then reading the SCDR. Reset clears the NF bit.

1 = Noise detected in SCDR
0 = No noise detected in SCDR

FE — Receiver Framing Error

This clearable, read-only flag is set when a logic zero is located where a stop bit should be in the character shifted into the receive shift register. If the received word causes both a framing error and an overrun error, the OR bit is set and the FE bit is not set. Clear the FE bit by reading the SCSR and then reading the SCDR. Reset clears the FE bit.

1 = Framing error
0 = No framing error

10.5.5 Baud Rate Register (Baud)

The baud rate register shown in Figure 10-10 selects the baud rate for both tne receiver and the transmitter.

		Bit 7	6	5	4	3	2	1	Bit 0
Baud $000D	Read:	▨	▨	SCP1	SCP0	▨	SCR2	SCR1	SCR0
	Write:								
	Reset:	U	U	0	0	U	U	U	U

▨ = Unimplemented U = Unaffected

Figure 10-10. Baud Rate Register (Baud)

SCP1 and SCP0 — SCI Prescaler Select Bits

These read/write bits control prescaling of the baud rate generator clock, as shown in Table 10-1. Resets clear both SCP1 and SCP0.

Table 10-1. Baud Rate Generator Clock Prescaling

SCP[1:0]	Baud Rate Generator Clock
00	Internal Clock ÷ 1
01	Internal Clock ÷ 3
10	Internal Clock ÷ 4
11	Internal Clock ÷ 13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate, as shown in Table 10-2. Reset has no effect on the SCR2–SCR0 bits.

Table 10-2. Baud Rate Selection

SCR[2:1:0]	SCI Baud Rate (Baud)
000	Prescaled Clock ÷ 1
001	Prescaled Clock ÷ 2
010	Prescaled Clock ÷ 4
011	Prescaled Clock ÷ 8
100	Prescaled Clock ÷ 16
101	Prescaled Clock ÷ 32
110	Prescaled Clock ÷ 64
111	Prescaled Clock ÷ 128

Table 10-3 shows all possible SCI baud rates derived from crystal frequencies of 2 MHz, 4 MHz, and 4.194304 MHz.

Table 10-3. Baud Rate Selection Examples

SCP[1:0]	SCR[2:1:0]	SCI Baud Rate		
		f_{osc} = 2 MHz	f_{osc} = 4 MHz	f_{osc} = 4.194304 MHz
00	000	62.50 kbaud	125 kbaud	131.1 kbaud
00	001	31.25 kbaud	62.50 kbaud	65.54 kbaud
00	010	15.63 kbaud	31.25 kbaud	32.77 kbaud
00	011	7813 Baud	15.63 kbaud	16.38 kbaud
00	100	3906 Baud	7813 Baud	8192 Baud
00	101	1953 Baud	3906 Baud	4096 Baud
00	110	976.6 Baud	1953 Baud	2048 Baud
00	111	488.3 Baud	976.6 Baud	1024 Baud
01	000	20.83 kbaud	41.67 kbaud	43.69 kbaud
01	001	10.42 kbaud	20.83 kbaud	21.85 kbaud
01	010	5208 Baud	10.42 kbaud	10.92 kbaud
01	011	2604 Baud	5208 Baud	5461 Baud
01	100	1302 Baud	2604 Baud	2731 Baud
01	101	651.0 Baud	1302 Baud	1365 Baud
01	110	325.5 Baud	651.0 Baud	682.7 Baud
01	111	162.8 Baud	325.5 Baud	341.3 Baud
10	000	15.63 kbaud	31.25 kbaud	32.77 kbaud
10	001	7813 Baud	15.63 kbaud	16.38 kbaud
10	010	3906 Baud	7813 Baud	8192 Baud
10	011	1953 Baud	3906 Baud	4906 Baud
10	100	976.6 Baud	1953 Baud	2048 Baud
10	101	488.3 Baud	976.6 Baud	1024 Baud
10	110	244.1 Baud	488.3 Baud	512.0 Baud
10	111	122.1 Baud	244.1 Baud	256.0 Baud
11	000	4808 Baud	9615 Baud	10.08 kbaud
11	001	2404 Baud	4808 Baud	5041 Baud
11	010	1202 Baud	2404 Baud	2521 Baud
11	011	601.0 Baud	1202 Baud	1260 Baud
11	100	300.5 Baud	601.0 Baud	630.2 Baud
11	101	150.2 Baud	300.5 Baud	315.1 Baud
11	110	75.12 Baud	150.2 Baud	157.5 Baud
11	111	37.56 Baud	75.12 Baud	78.77 Baud

SECTION 11
SERIAL PERIPHERAL INTERFACE (SPI)

11.1 Introduction

The SPI module allows full-duplex, synchronous, serial communication with peripheral devices.

11.2 Features

Features of the SPI include the following:

- Full Duplex Operation
- Master and Slave Modes
- Four Programmable Master Mode Frequencies (1.05 MHz Maximum)
- 2.1 MHz Maximum Slave Mode Frequency
- Serial Clock with Programmable Polarity and Phase
- End of Transmission Interrupt Flag
- Write Collision Error Flag
- Bus Contention Error Flag

Figure 11-1 shows the structure of the SPI module. Figure 11-2 is a summary of the SPI I/O registers.

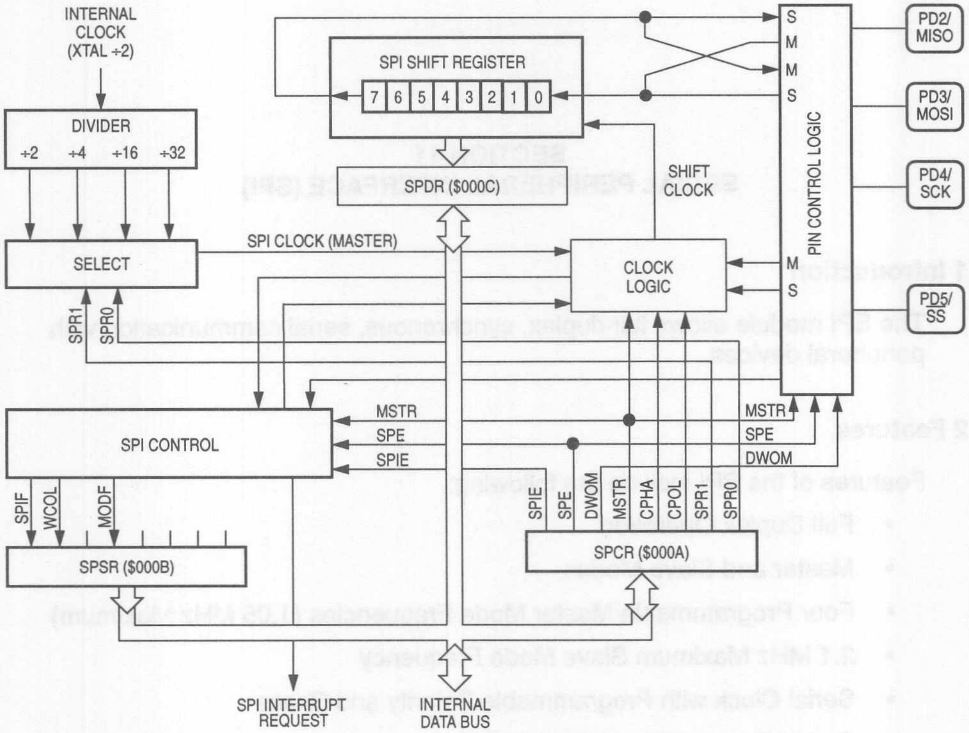


Figure 11-1. SPI Block Diagram

Register Name	Bit 7	6	5	4	3	2	1	Bit 0	Addr.
SPI Control Register (SPCR)	SPIE	SPE	0	MSTR	CPOL	CPHA	SPR1	SPR0	$000A
SPI Status Register (SPSR)	SPIF	WCOL	0	MODF	▨	▨	▨	▨	$000B
SPI Data Register (SPDR)	Bit 7	6	5	4	3	2	1	Bit 0	$000C

▨ = Unimplemented

Figure 11-2. SPI I/O Register Summary

11.3 Operation

The master/slave SPI allows full-duplex, synchronous, serial communication between the MCU and peripheral devices, including other MCUs. As the 8-bit shift register of a master SPI transmits each byte to another device, a byte from the receiving device enters the master SPI shift register. A clock signal from the master SPI synchronizes data transmission.

Only a master SPI can initiate transmissions. Software begins the transmission from a master SPI by writing to the SPI data register (SPDR). The SPDR does not buffer data being transmitted from the SPI. Data written to the SPDR goes directly into the shift register and begins the transmission immediately under the control of the serial clock. The transmission ends after eight cycles of the serial clock when the SPI flag (SPIF) becomes set. At the same time that SPIF becomes set, the data shifted into the master SPI from the receiving device transfers to the SPDR. The SPDR buffers data being received by the SPI. Before the master SPI sends the next byte, software must clear the SPIF bit by reading the SPSR and then accessing the SPDR.

In a slave SPI, data enters the shift register under the control of the serial clock from the master SPI. After a byte enters the shift register of a slave SPI, it transfers to the SPDR. To prevent an overrun condition, slave software must then read the byte in the SPDR before another byte enters the shift register and is ready to transfer to the SPDR.

Figure 11-3 shows how a master SPI exchanges data with a slave SPI.

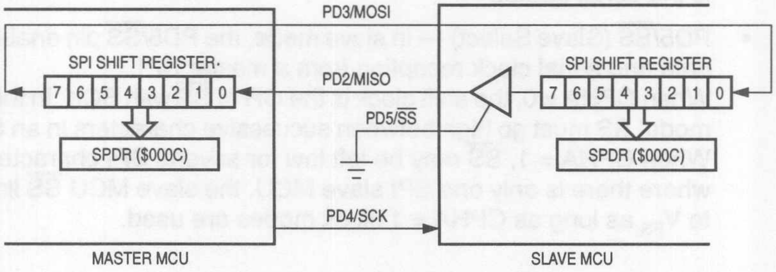


Figure 11-3. Master/Slave Connections

11.3.1 Pin Functions in Master Mode

Setting the MSTR bit in the SPI control register (SPCR) configures the SPI for operation in master mode. The master-mode functions of the SPI pins are:

- PD4/SCK (Serial Clock) — In master mode, the PD4/SCK pin is the synchronizing clock output.
- PD3/MOSI (Master Output, Slave Input) — In master mode, the PD3/MOSI pin is the serial output.
- PD2/MISO (Master Input, Slave Output) — In master mode, the PD2/MISO pin is configured as the serial input.
- PD5/$\overline{\text{SS}}$ (Slave Select) — In master mode, the PD5/$\overline{\text{SS}}$ pin protects against driver contention caused by the simultaneous operation of two SPIs in master mode. A logic zero on the PD5/$\overline{\text{SS}}$ pin of a master SPI disables the SPI, clears the MSTR bit, and sets the mode-fault flag (MODF).

11.3.2 Pin Functions in Slave Mode

Clearing the MSTR bit in the SPCR configures the SPI for operation in slave mode. The slave-mode functions of the SPI pins are:

- PD4/SCK (Serial Clock) — In slave mode, the PD4/SCK pin is the input for the synchronizing clock signal from the master SPI.
- PD3/MOSI (Master Output, Slave Input) — In slave mode, the PD3/MOSI pin is the serial input.
- PD2/MISO (Master Input, Slave Output) — In slave mode, the PD2/MISO pin is the serial output.
- PD5/$\overline{\text{SS}}$ (Slave Select) — In slave mode, the PD5/$\overline{\text{SS}}$ pin enables the SPI for data and serial clock reception from a master SPI.
  When CPHA = 0, the shift clock is the OR of $\overline{\text{SS}}$ with SCK. In this clock phase mode, SS must go high between successive characters in an SPI message. When CPHA = 1, $\overline{\text{SS}}$ may be left low for several SPI characters. In cases where there is only one SPI slave MCU, the slave MCU $\overline{\text{SS}}$ line can be tied to V_{SS} as long as CPHA = 1 clock modes are used.

11.4 Multiple-SPI Systems

In a multiple-SPI system, all PD4/SCK pins are connected together, all PD3/MOSI pins are connected together, and all PD2/MISO pins are connected together.

Before a transmission, one SPI is configured as master and the rest are configured as slaves. Figure 11-4 is a block diagram showing a single master SPI and three slave SPIs.

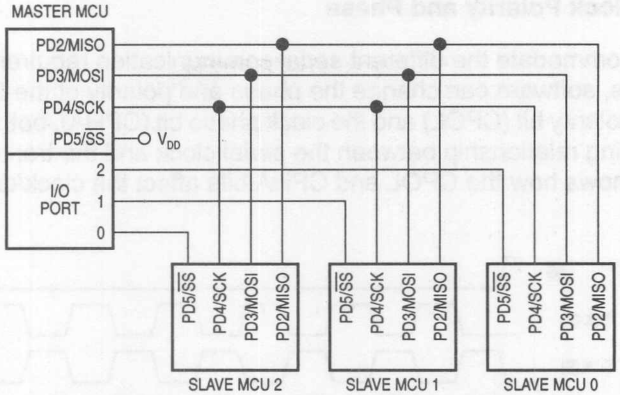


Figure 11-4. One Master and Three Slaves Block Diagram

Figure 11-5 is another block diagram with two master/slave SPIs and three slave SPIs.

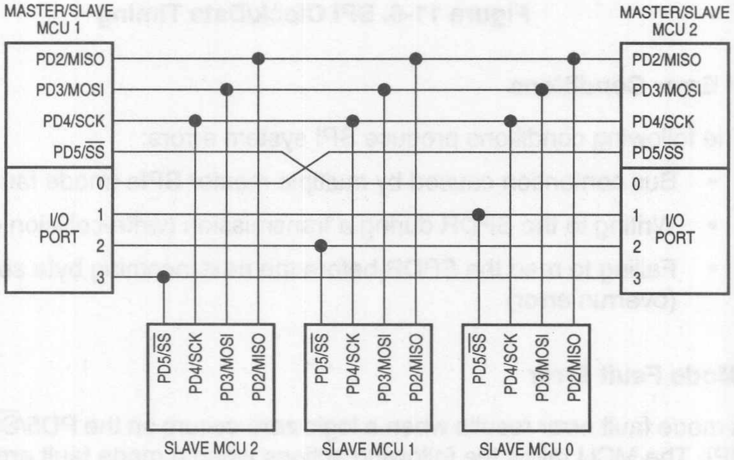


Figure 11-5. Two Master/Slaves and Three Slaves Block Diagram

11.5 Serial Clock Polarity and Phase

To accommodate the different serial communication requirements of peripheral devices, software can change the phase and polarity of the SPI serial clock. The clock polarity bit (CPOL) and the clock phase bit (CPHA), both in the SPCR, control the timing relationship between the serial clock and the transmitted data. Figure 11-6 shows how the CPOL and CPHA bits affect the clock/data timing.

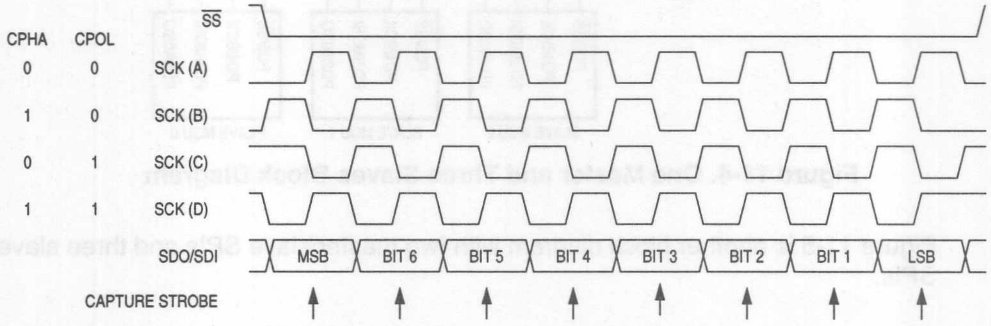


Figure 11-6. SPI Clock/Data Timing

11.6 SPI Error Conditions

The following conditions produce SPI system errors:

- Bus contention caused by multiple master SPIs (mode fault error)
- Writing to the SPDR during a transmission (write-collision error)
- Failing to read the SPDR before the next incoming byte sets the SPIF bit (overrun error)

11.6.1 Mode Fault Error

A mode fault error results when a logic zero occurs on the PD5/$\overline{\text{SS}}$ pin of a master SPI. The MCU takes the following actions when a mode fault error occurs:

- Puts the SPI in slave mode by clearing the MSTR bit
- Disables the SPI by clearing the SPE bit
- Sets the MODF bit

11.6.2 Write Collision Error

Writing to the SPDR during a transmission causes a write collision error and sets the WCOL bit in the SPSR. Either a master SPI or a slave SPI can generate a write collision error.

- Master — A master SPI can cause a write collision error by writing to the SPDR while the previously written byte is still being shifted out to the PD3/MOSI pin. The error does not affect the transmission of the previously written byte, but the byte that caused the error is lost.
- Slave — A slave SPI can cause a write collision error in either of two ways, depending on the state of the CPHA bit:
  - CPHA = 0 — A slave SPI can cause a write collision error by writing to the SPDR while the PD5/$\overline{SS}$ pin is at logic zero. The error does not affect the byte in the SPDR, but the byte that caused the error is lost.
  - CPHA = 1 — A slave SPI can cause a write collision error by writing to the SPDR while receiving a transmission, that is, between the first active SCK edge and the end of the eighth SCK cycle. The error does not affect the transmission from the master SPI, but the byte that caused the error is lost.

11.6.3 Overrun Error

Failing to read the byte in the SPDR before a subsequent byte enters the shift register causes an overrun condition. In an overrun condition, all incoming data is lost until software clears SPIF. The overrun condition has no flag.

11.7 SPI Interrupts

The SPIF bit in the SPSR indicates a byte has shifted into or out of the SPDR. The SPIF bit is a source of SPI interrupt requests. The SPI interrupt enable bit (SPIE) in the SPCR is the local mask for SPIF interrupts.

The MODF bit in the SPSR indicates a mode error and is a source of SPI interrupt requests. The MODF bit is set when a logic zero occurs on the PD5/$\overline{SS}$ pin while the MSTR bit is set. The SPI interrupt enable bit (SPIE) in the SPCR is the local mask for MODF interrupts.

11.8 SPI I/O Registers

The following I/O registers control and monitor SPI operation:

- SPI Data Register (SPDR)
- SPI Control Register (SPCR)
- SPI Status Register (SPSR)

11.8.1 SPI Data Register (SPDR)

The SPDR shown in Figure 11-7 is the read buffer for characters received by the SPI. Writing a byte to the SPDR places the byte directly into the SPI shift register.

		Bit 7	6	5	4	3	2	1	Bit 0
SPDR $000C	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0
	Reset:	Unaffected by reset							

Figure 11-7. SPI Data Register (SPDR)

11.8.2 SPI Control Register (SPCR)

The SPCR shown in Figure 11-8 has the following functions:

- Enables SPI interrupt requests
- Enables the SPI
- Configures the SPI as master or slave
- Selects serial clock polarity, phase, and frequency

		Bit 7	6	5	4	3	2	1	Bit 0
SPCR $000A	Read: Write:	SPIE	SPE	▨	MSTR	CPOL	CPHA	SPR1	SPR0
	Reset:	0	0		0	U	U	U	U

▨ = Unimplemented U = Unaffected

Figure 11-8. SPI Control Register (SPCR)

SPIE — SPI Interrupt Enable Bit

This read/write bit enables SPI interrupts. Reset clears the SPIE bit.
1 = SPI interrupts enabled
0 = SPI interrupts disabled

SPI — SPI Enable Bit

This read/write bit enables the SPI. Reset clears the SPE bit.
1 = SPI enabled
0 = SPI disabled

MSTR — Master Bit

This read/write bit selects master mode operation or slave mode operation. Reset clears the MSTR bit.

1 = Master mode
0 = Slave mode

CPOL — Clock Polarity Bit

This read/write bit determines the logic state of the PD4/SCK pin between transmissions. To transmit data between SPIs, the SPIs must have identical CPOL bits. Reset has no effect on the CPOL bit.

1 = PD4/SCK pin at logic one between transmissions
0 = PD4/SCK pin at logic zero between transmissions

CPHA — Clock Phase Bit

This read/write bit controls the timing relationship between the serial clock and SPI data. To transmit data between SPIs, the SPIs must have identical CPHA bits. When CPHA = 0, the PD5/$\overline{\text{SS}}$ pin of the slave SPI must be set to logic one between bytes. Reset has no effect on the CPHA bit.

1 = Edge following first active edge on PD4/SCK latches data
0 = First active edge on PD4/SCK latches data

SPR1 and SPR0 — SPI Clock Rate Bits

These read/write bits select the master mode serial clock rate, as shown in Table 11-1. The SPR1 and SPR0 bits of a slave SPI have no effect on the serial clock. Reset has no effect on SPR1 and SPR0.

Table 11-1. SPI Clock Rate Selection

SPR[1:0]	SPI Clock Rate
00	Internal Clock ÷ 2
01	Internal Clock ÷ 4
10	Internal Clock ÷ 16
11	Internal Clock ÷ 32

11.8.3 SPI Status Register (SPSR)

The SPSR shown in Figure 11-9 contains flags to signal the following conditions:

- SPI transmission complete
- Write collision
- Mode fault

SPSR $000B		Bit 7	6	5	4	3	2	1	Bit 0
	Read:	SPIF	WCOL	▨	MODF	▨	▨	▨	▨
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
	Reset:	0	0		0				

▨ = Unimplemented

Figure 11-9. SPI Status Register (SPSR)

SPIF — SPI Flag Bit

This clearable, read-only bit is set each time a byte shifts out of or into the shift register. SPIF generates an interrupt request if the SPIE bit in the SPCR is also set. Clear SPIF by reading the SPSR with SPIF set, and then reading or writing the SPDR. Reset clears the SPIF bit.

1 = Transmission complete
0 = Transmission not complete

WCOL — Write Collision Bit

This clearable, read-only flag is set when software writes to the SPDR while a transmission is in progress. Clear the WCOL bit by reading the SPSR with WCOL set, and then reading or writing the SPDR. Reset clears WCOL.

1 = Invalid write to SPDR
0 = No invalid write to SPDR

MODF — Mode Fault Bit

This clearable, read-only bit is set when a logic zero occurs on the PD5/$\overline{SS}$ pin while the MSTR bit is set. MODF generates an interrupt request if the SPIE bit is also set. Clear the MODF bit by reading the SPSR with MODF set and then writing to the SPCR. Reset clears MODF.

1 = PD5/$\overline{SS}$ pulled low while MSTR bit set
0 = PD5/$\overline{SS}$ not pulled low while MSTR bit set

SECTION 12
INSTRUCTION SET

12.1 Introduction

This section describes the addressing modes and instruction types.

12.2 Addressing Modes

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes define the manner in which the CPU finds the data required to execute an instruction. The eight addressing modes are:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, no offset
- Indexed, 8-bit offset
- Indexed, 16-bit offset
- Relative

12.2.1 Inherent

Inherent instructions are those that have no operand, such as return from interrupt (RTI) and stop (STOP). Some of the inherent instructions act on data in the CPU registers, such as set carry flag (SEC) and increment accumulator (INCA). Inherent instructions require no operand address and are one byte long.

12.2.2 Immediate

Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no operand address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

12.2.3 Direct

Direct instructions can access any of the first 256 memory addresses with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address.

12.2.4 Extended

Extended instructions use three bytes and can access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.

When using the Motorola assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

12.2.5 Indexed, No Offset

Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.

12.2.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are 2-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the effective address of the operand. These instructions can access locations $0000–$01FE.

Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 ($01FE). The k value is typically in the index register, and the address of the beginning of the table is in the byte following the opcode.

12.2.7 Indexed, 16-Bit Offset

Indexed, 16-bit offset instructions are 3-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the effective address of the operand. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset.

Indexed, 16-bit offset instructions are useful for selecting the kth element in an n-element table anywhere in memory.

As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

12.2.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the effective branch destination by adding the signed byte following the opcode to the contents of the program counter. If the branch condition is not true, the CPU goes to the next instruction. The offset is a signed, two's complement byte that gives a branching range of –128 to +127 bytes from the address of the next location after the branch instruction.

When using the Motorola assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and verifies that it is within the span of the branch.

12.3 Instruction Types

The MCU instructions fall into the following five categories:

- Register/memory instructions
- Read-modify-write instructions
- Jump/branch instructions
- Bit manipulation instructions
- Control instructions

12.3.1 Register/Memory Instructions

These instructions operate on CPU registers and memory locations. Most of them use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory.

Table 12-1. Register/Memory Instructions

Instruction	Mnemonic
Add Memory Byte and Carry Bit to Accumulator	ADC
Add Memory Byte to Accumulator	ADD
AND Memory Byte with Accumulator	AND
Bit Test Accumulator	BIT
Compare Accumulator	CMP
Compare Index Register with Memory Byte	CPX
EXCLUSIVE OR Accumulator with Memory Byte	EOR
Unconditional Jump	JMP
Jump to Subroutine	JSR
Load Accumulator with Memory Byte	LDA
Load Index Register with Memory Byte	LDX
Multiply	MUL
OR Accumulator with Memory Byte	ORA
Subtract Memory Byte and Carry Bit from Accumulator	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract Memory Byte from Accumulator	SUB

12.3.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register.

NOTE

Do not use read-modify-write instructions on write-only registers.

Table 12-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left (Same as LSL)	ASL
Arithmetic Shift Right	ASR
Clear Bit	BCLR
Set Bit	BSET
Clear Register	CLR
Complement (One's Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left (Same as ASL)	LSL
Logical Shift Right	LSR
Negate (Two's Complement)	NEG
Rotate Left through Carry Bit	ROL
Rotate Right through Carry Bit	ROR
Test for Negative or Zero	TST[1]

NOTES:

[1] TST is an exception to the read-modify-write sequence because it does not write a replacement value.

12.3.3 Jump/Branch Instructions

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The unconditional jump instruction (JMP) and the jump to subroutine instruction (JSR) have no register operand. Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed.

The BRCLR and BRSET instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These 3-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the effective branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register.

Table 12-3. Jump and Branch Instructions

Instruction	Mnemonic
Branch if Carry Bit Clear	BCC
Branch if Carry Bit Set	BCS
Branch if Equal	BEQ
Branch if Half-Carry Bit Clear	BHCC
Branch if Half-Carry Bit Set	BHCS
Branch if Higher	BHI
Branch if Higher or Same	BHS
Branch if $\overline{\text{IRQ}}$ Pin High	BIH
Branch if $\overline{\text{IRQ}}$ Pin Low	BIL
Branch if Lower	BLO
Branch if Lower or Same	BLS
Branch if Interrupt Mask Clear	BMC
Branch if Minus	BMI
Branch if Interrupt Mask Set	BMS
Branch if Not Equal	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit Clear	BRCLR
Branch Never	BRN
Branch if Bit Set	BRSET
Branch to Subroutine	BSR
Unconditional Jump	JMP
Jump to Subroutine	JSR

12.3.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port registers, port data direction registers, timer registers, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations.

Table 12-4. Bit Manipulation Instructions

Instruction	Mnemonic
Bit Clear	BCLR
Branch if Bit Clear	BRCLR
Branch if Bit Set	BRSET
Bit Set	BSET

NOTE

Avoid using bit clear and bit set instructions in registers containing write-only bits, which may be changed inadvertently.

12.3.5 Control Instructions

These instructions act on CPU registers and control CPU operation during program execution.

Table 12-5. Control Instructions

Instruction	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
No Operation	NOP
Reset Stack Pointer	RSP
Return from Interrupt	RTI
Return from Subroutine	RTS
Set Carry Bit	SEC
Set Interrupt Mask	SEI
Stop Oscillator and Enable $\overline{IRQ}$ Pin	STOP
Software Interrupt	SWI
Transfer Accumulator to Index Register	TAX
Transfer Index Register to Accumulator	TXA
Stop CPU Clock and Enable Interrupts	WAIT

12.4 Instruction Set Summary

Table 12-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 12-6. Instruction Set Summary

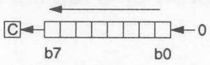
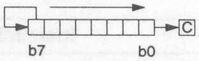
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X	Add without Carry	A ← (A) + (M)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND #*opr* AND *opr* AND *opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	↕	↕	↕	DIR INH INH IX1 IX	37 47 57 67 77	dd ff	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff 	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if Bit n clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2
CLR *opr*	Clear Byte	M ← $00	—	—	0	1	—	DIR	3F	dd	5
CLRA		A ← $00						INH	4F		3
CLRX		X ← $00						INH	5F		3
CLR *opr*,X		M ← $00						IX1	6F	ff	6
CLR ,X		M ← $00						IX	7F		5
CMP #*opr*	Compare Accumulator with Memory Byte	(A) – (M)	—	—	↕	↕	↕	IMM	A1	ii	2
CMP *opr*								DIR	B1	dd	3
CMP *opr*								EXT	C1	hh ll	4
CMP *opr*,X								IX2	D1	ee ff	5
CMP *opr*,X								IX1	E1	ff	4
CMP ,X								IX	F1		3
COM *opr*	Complement Byte (One's Complement)	$M \leftarrow (\overline{M}) = \$FF - (M)$	—	—	↕	↕	1	DIR	33	dd	5
COMA		$A \leftarrow (\overline{A}) = \$FF - (A)$						INH	43		3
COMX		$X \leftarrow (\overline{X}) = \$FF - (X)$						INH	53		3
COM *opr*,X		$M \leftarrow (\overline{M}) = \$FF - (M)$						IX1	63	ff	6
COM ,X		$M \leftarrow (\overline{M}) = \$FF - (M)$						IX	73		5
CPX #*opr*	Compare Index Register with Memory Byte	(X) – (M)	—	—	↕	↕	↕	IMM	A3	ii	2
CPX *opr*								DIR	B3	dd	3
CPX *opr*								EXT	C3	hh ll	4
CPX *opr*,X								IX2	D3	ee ff	5
CPX *opr*,X								IX1	E3	ff	4
CPX ,X								IX	F3		3
DEC *opr*	Decrement Byte	M ← (M) – 1	—	—	↕	↕	—	DIR	3A	dd	5
DECA		A ← (A) – 1						INH	4A		3
DECX		X ← (X) – 1						INH	5A		3
DEC *opr*,X		M ← (M) – 1						IX1	6A	ff	6
DEC ,X		M ← (M) – 1						IX	7A		5
EOR #*opr*	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	↕	↕	—	IMM	A8	ii	2
EOR *opr*								DIR	B8	dd	3
EOR *opr*								EXT	C8	hh ll	4
EOR *opr*,X								IX2	D8	ee ff	5
EOR *opr*,X								IX1	E8	ff	4
EOR ,X								IX	F8		3
INC *opr*	Increment Byte	M ← (M) + 1	—	—	↕	↕	—	DIR	3C	dd	5
INCA		A ← (A) + 1						INH	4C		3
INCX		X ← (X) + 1						INH	5C		3
INC *opr*,X		M ← (M) + 1						IX1	6C	ff	6
INC ,X		M ← (M) + 1						IX	7C		5
JMP *opr*	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR	BC	dd	2
JMP *opr*								EXT	CC	hh ll	3
JMP *opr*,X								IX2	DC	ee ff	4
JMP *opr*,X								IX1	EC	ff	3
JMP ,X								IX	FC		2

Table 12-6. Instruction Set Summary (Continued)

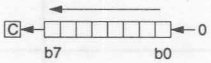
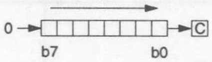
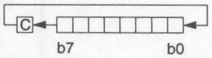
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address						DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	↕	↕	DIR INH INH IX1 IX	34 44 54 64 74	dd ff	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		11
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = \$00 – (M) A ← –(A) = \$00 – (A) X ← –(X) = \$00 – (X) M ← –(M) = \$00 – (M) M ← –(M) = \$00 – (M)	—	—	↕	↕	↕	DIR INH INH IX1 IX	30 40 50 60 70	dd ff	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff	5 3 3 6 5

Table 12-6. Instruction Set Summary (Continued)

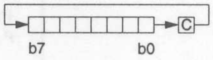
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff 	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	INH	80		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff 	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff 	4 5 6 5 4
STOP	Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff 	4 5 6 5 4
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff 	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
TST *opr*	Test Memory Byte for Negative or Zero	(M) – $00	—	—	↕	↕	—	DIR	3D	dd	4
TSTA								INH	4D		3
TSTX								INH	5D		3
TST *opr*,X								IX1	6D	ff	5
TST ,X								IX	7D		4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	0	—	—	—	INH	8F		2

A	Accumulator
C	Carry/borrow flag
CCR	Condition code register
dd	Direct address of operand
dd rr	Direct address of operand and relative offset of branch instruction
DIR	Direct addressing mode
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing
EXT	Extended addressing mode
ff	Offset byte in indexed, 8-bit offset addressing
H	Half-carry flag
hh ll	High and low bytes of operand address in extended addressing
I	Interrupt mask
ii	Immediate operand byte
IMM	Immediate addressing mode
INH	Inherent addressing mode
IX	Indexed, no offset addressing mode
IX1	Indexed, 8-bit offset addressing mode
IX2	Indexed, 16-bit offset addressing mode
M	Memory location
N	Negative flag
n	Any bit
opr	Operand (one or two bytes)
PC	Program counter
PCH	Program counter high byte
PCL	Program counter low byte
REL	Relative addressing mode
rel	Relative program counter offset byte
rr	Relative program counter offset byte
SP	Stack pointer
X	Index register
Z	Zero flag
#	Immediate value
∧	Logical AND
∨	Logical OR
⊕	Logical EXCLUSIVE OR
( )	Contents of
–( )	Negation (two's complement)
←	Loaded with
?	If
:	Concatenated with
↕	Set or cleared
—	Not affected

Table 12-7. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	DIR	DIR	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
MSB LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSB LSB
0	5 BRSET0 3 DIR	5 BSET0 2 DIR	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0
1	5 BRCLR0 3 DIR	5 BCLR0 2 DIR	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1
2	5 BRSET1 3 DIR	5 BSET1 2 DIR	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2
3	5 BRCLR1 3 DIR	5 BCLR1 2 DIR	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3
4	5 BRSET2 3 DIR	5 BSET2 2 DIR	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4
5	5 BRCLR2 3 DIR	5 BCLR2 2 DIR	3 BCS/BLO 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5
6	5 BRSET3 3 DIR	5 BSET3 2 DIR	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6
7	5 BRCLR3 3 DIR	5 BCLR3 2 DIR	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7
8	5 BRSET4 3 DIR	5 BSET4 2 DIR	3 BHCC 2 REL	5 ASL/LSL 2 DIR	3 ASLA/LSLA 1 INH	3 ASLX/LSLX 1 INH	6 ASL/LSL 2 IX1	5 ASL/LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8
9	5 BRCLR4 3 DIR	5 BCLR4 2 DIR	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9
A	5 BRSET5 3 DIR	5 BSET5 2 DIR	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A
B	5 BRCLR5 3 DIR	5 BCLR5 2 DIR	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B
C	5 BRSET6 3 DIR	5 BSET6 2 DIR	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C
D	5 BRCLR6 3 DIR	5 BCLR6 2 DIR	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D
E	5 BRSET7 3 DIR	5 BSET7 2 DIR	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E
F	5 BRCLR7 3 DIR	5 BCLR7 2 DIR	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F

INH = Inherent
IMM = Immediate
DIR = Direct
EXT = Extended

REL = Relative
IX = Indexed, No Offset
IX1 = Indexed, 8-Bit Offset
IX2 = Indexed, 16-Bit Offset

	MSB LSB	0	MSB of Opcode in Hexadecimal
LSB of Opcode in Hexadecimal	0	5 BRSET0 3 DIR	Number of Cycles Opcode Mnemonic Number of Bytes/Addressing Mode

SECTION 13
ELECTRICAL SPECIFICATIONS

13.1 Introduction

This section contains electrical and timing specifications.

13.2 Maximum Ratings

Maximum ratings are the extreme limits the device can be exposed to without causing permanent damage to the chip. The MCU contains circuitry that protects the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in Table 13-1. Keep V_{IN} and V_{OUT} within the range $V_{SS} \leq (V_{IN}$ or $V_{OUT}) \leq V_{DD}$. Connect unused inputs to appropriate voltage level, either V_{SS} or V_{DD}.

Table 13-1. Maximum Ratings[1]

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	– 0.3 to + 7.0	V
Input Voltage	V_{IN}	V_{SS} – 0.3 to V_{DD} + 0.3	V
Programming Voltage	V_{PP}	V_{DD} – 0.3 to 16.0	V
Bootstrap Mode ($\overline{\text{IRQ}}$ Pin Only)	V_{IN}	V_{SS} – 0.3 to 2 x V_{DD} + 0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Storage Temperature Range	T_{STG}	– 65 to + 150	°C

1. Voltages referenced to V_{SS}

13.3 Operating Temperature Range

Table 13-2. Operating Temperature Range[1]

Rating	Symbol	Value	Unit
Operating Temperature Range[2] MC68HC705C8ACB, CFB, CP, CFN	T_A	T_L to T_H – 40 to + 85	°C

1. Voltages referenced to V_{SS}
2. C = Extended temperature range (– 40 to + 85 °C) P = Plastic dual in-line package (PDIP)
   B = Plastic shrink dual in-line package (SDIP) FN = Plastic-leaded chip carrier (PLCC)
   FB = Quad flat pack (QFP)

13.4 Thermal Characteristics

Table 13-3. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Plastic Dual In-Line Package (DIP)	θ_{JA}	60	°C/W
Plastic Leaded Chip Carrier (PLCC)		70	
Quad Flat Pack (QFP)		95	
Plastic Shrink DIP (SDIP)		60	

13.5 Power Considerations

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{DD} \times V_{DD}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications, $P_{I/O} < P_{INT}$ and can be neglected.

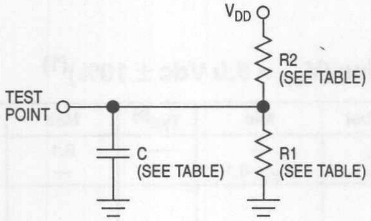
The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273\ °C) \qquad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D (T_A + 273\ °C) + \theta_{JA} \bullet P_D^2 \qquad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.



V_{DD} = 4.5 V

Pins	R1	R2	C
PA7–PA0 PB7–PB0 PC7–PC0 PD4–PD1	3.26 kΩ	2.38 kΩ	50 pF
PD7, PD5, PD0	1.9 kΩ	2.26 kΩ	200 pF

V_{DD} = 3.0 V

Pins	R1	R2	C
PA7–PA0 PB7–PB0 PC7–PC0 PD4–PD1	10.91kΩ	6.32 kΩ	50 pF
PD7, PD5, PD0	6 kΩ	6 kΩ	200 pF

Figure 13-1. Equivalent Test Load

13.6 DC Electrical Characteristics

Table 13-4. DC Electrical Characteristics (V_{DD} = 5.0 Vdc ± 10%)[1]

Characteristic	Symbol	Min	Typ[2]	Max	Unit
Output Voltage, $I_{Load} \leq$ 10.0 µA	V_{OL} V_{OH}	— V_{DD} – 0.1	— —	0.1 —	V
Output High Voltage I_{LOAD} = –0.8 mA PA7–PA0, PB7–PB0, PC6–PC0, TCMP (See Figure 13-2.) I_{LOAD} = –1.6 mA PD4–PD1 (See Figure 13-3.) I_{LOAD} –5.0 mA PC7	V_{OH}	V_{DD} – 0.8	— — —	— — —	V
Output Low Voltage (See Figure 13-4.) I_{LOAD} = 1.6 mA PA7–PA0, PB7–PB0, PC6–PC0, PD4–PD1 I_{LOAD} = 20 mA PC7	V_{OL}	— —	— —	0.4 0.4	V
Input High Voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
EPROM Programming Voltage	V_{PP}	14.5	14.75	15.0	V
EPROM/OTPROM Programming Current	I_{PP}	—	5	10	mA
User Mode Current	I_{PP}	—	—	± 10	mA
Data-Retention Mode (0° to 70 °C)	V_{RM}	2.0	—	—	V
Supply Current [3] Run[4] WAIT[5] STOP[6] 25 °C –40 °C to +85 °C	I_{DD}	— — — —	5.0 1.95 5.0 5.0	7.0 3.0 50 50	mA mA µA µA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7–PB0, PC7–PC0, PD4–PD1, PD7, $\overline{RESET}$	I_{IL}	—	—	± 10	µA
Input Current $\overline{IRQ}$, TCAP, OSC1, PD0, PD5	I_{IN}	—	—	± 1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, PD0–PD5, PD7	C_{OUT} C_{IN}	— —	— —	12 8	pF

NOTES:

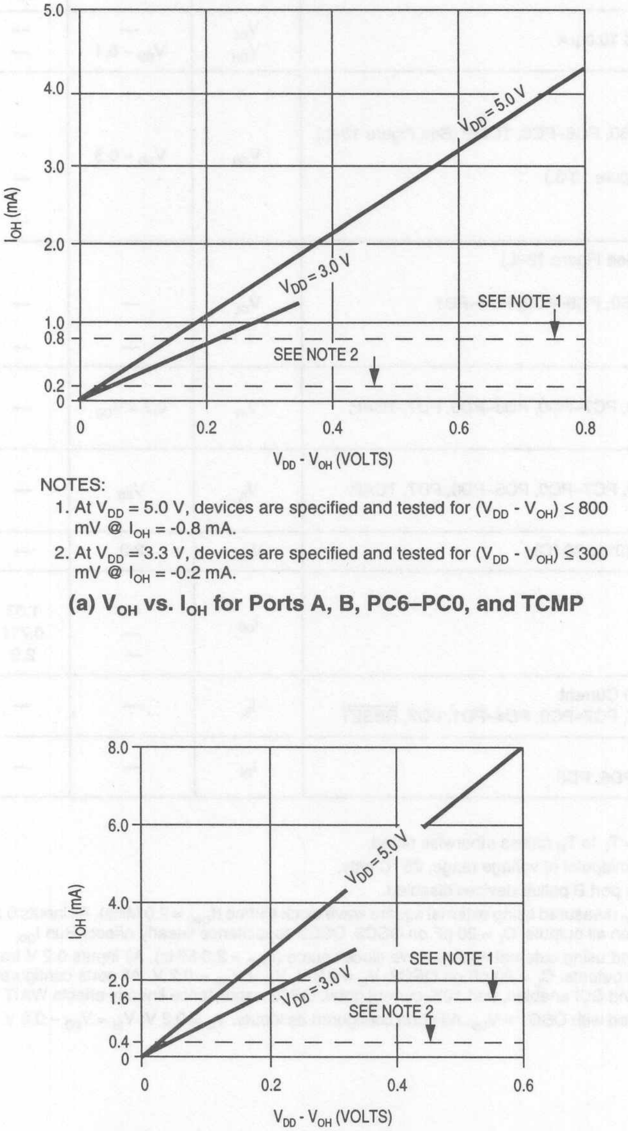
1. V_{DD} = 5 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range at 25 °C.
3. I_{DD} measured with port B pullup devices disabled.
4. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. OSC2 capacitance linearly affects Run I_{DD}.
5. WAIT I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V. All ports configured as inputs. SPI and SCI disabled. If SPI and SCI enabled, add 10% current draw. OSC2 capacitance linearly affects WAIT I_{DD}.
6. STOP I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs. V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V.

Table 13-5. DC Electrical Characteristics (V_{DD} = 3.3 Vdc ± 10%)[1]

Characteristic	Symbol	Min	Typ[2]	Max	Unit
Output Voltage, $I_{Load} \le 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage $I_{LOAD} = -0.2$ mA PA7–PA0, PB7–PB0, PC6–PC0, TCMP (See Figure 13-2.) $I_{LOAD} = -0.4$ mA PD4–PD1 (See Figure 13-3.) $I_{LOAD} = -1.5$ mA PC7	V_{OH}	$V_{DD} - 0.3$	— — —	— — —	V
Output Low Voltage (See Figure 13-4.) $I_{LOAD} = 0.4$ mA PA7–PA0, PB7–PB0, PC6–PC0, PD4–PD1 $I_{LOAD} = 6.0$ mA PC7	V_{OL}	— —	— —	0.3 0.3	V
Input High Voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB7–PB0, PC7–PC0, PD5–PD0, PD7, TCAP, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSCI	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Data-Retention Mode (0° to 70 °C)	V_{RM}	2.0	—	—	V
Supply Current[3] Run[4] WAIT[5] STOP[6]	I_{DD}	— — —	1.53 0.711 2.0	3.0 1.0 20	mA mA µA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7–PB0, PC7–PC0, PD4–PD1, PD7, $\overline{\text{RESET}}$	I_{IL}	—	—	± 10	µA
Input Current $\overline{\text{IRQ}}$, TCAP, OSC1, PD5, PD0	I_{IN}	—	—	± 1	µA

NOTES:

1. V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted.
2. Typical values at midpoint of voltage range, 25 °C only.
3. I_{DD} measured with port B pullup devices disabled.
4. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 2.0 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. OSC2 capacitance linearly affects Run I_{DD}.
5. WAIT I_{DD} measured using external square wave clock source (f_{OSC} = 2.0 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V. All ports configured as inputs. SPI and SCI disabled. If SPI and SCI enabled, add 10% current draw. OSC2 capacitance linearly affects WAIT I_{DD}.
6. STOP I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs. V_{IL} = 0.2 V; $V_{IH} = V_{DD} - 0.2$ V.



NOTES:

1. At V_{DD} = 5.0 V, devices are specified and tested for (V_{DD} - V_{OH}) ≤ 800 mV @ I_{OH} = -0.8 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for (V_{DD} - V_{OH}) ≤ 300 mV @ I_{OH} = -0.2 mA.

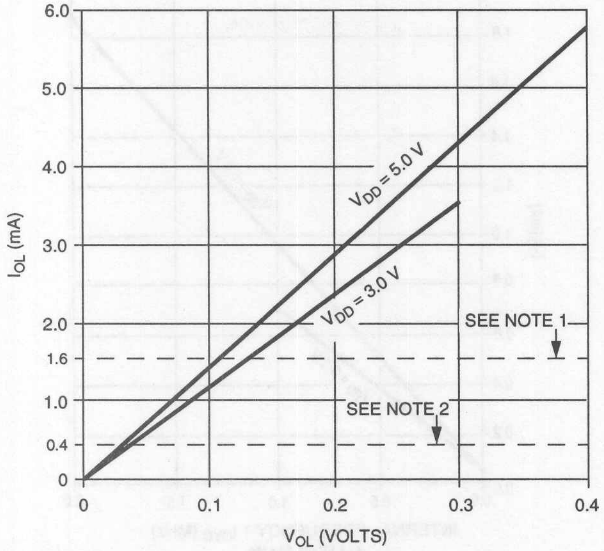
(a) V_{OH} vs. I_{OH} for Ports A, B, PC6–PC0, and TCMP

NOTES:

1. At V_{DD} = 5.0 V, devices are specified and tested for (V_{DD} - V_{OH}) ≤ 800 mV @ I_{OH} = -1.6 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for (V_{DD} - V_{OH}) ≤ 300 mV @ I_{OH} = -0.4 mA.

(b) V_{OH} vs. I_{OH} for PD4–PD1

Figure 13-2. Typical Voltage Compared to Current

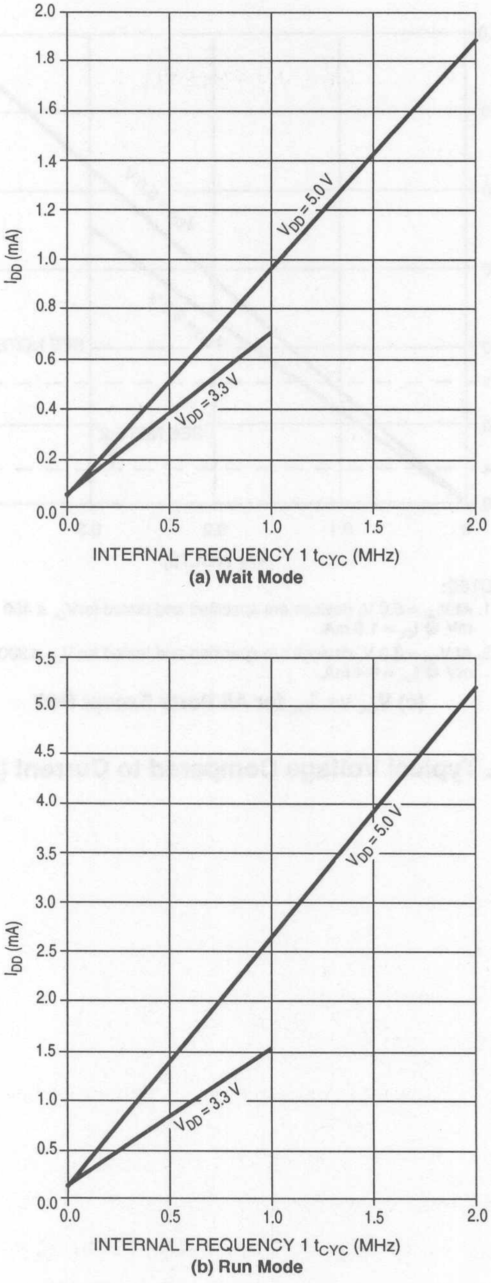


NOTES:

1. At V_{DD} = 5.0 V, devices are specified and tested for$V_{OL} \leq$ 400 mV @ I_{OL} = 1.6 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for $V_{OL} \leq$ 300 mV @ I_{OL} = 0.4 mA.

(c) V_{OL} vs. I_{OL} for All Ports Except PC7

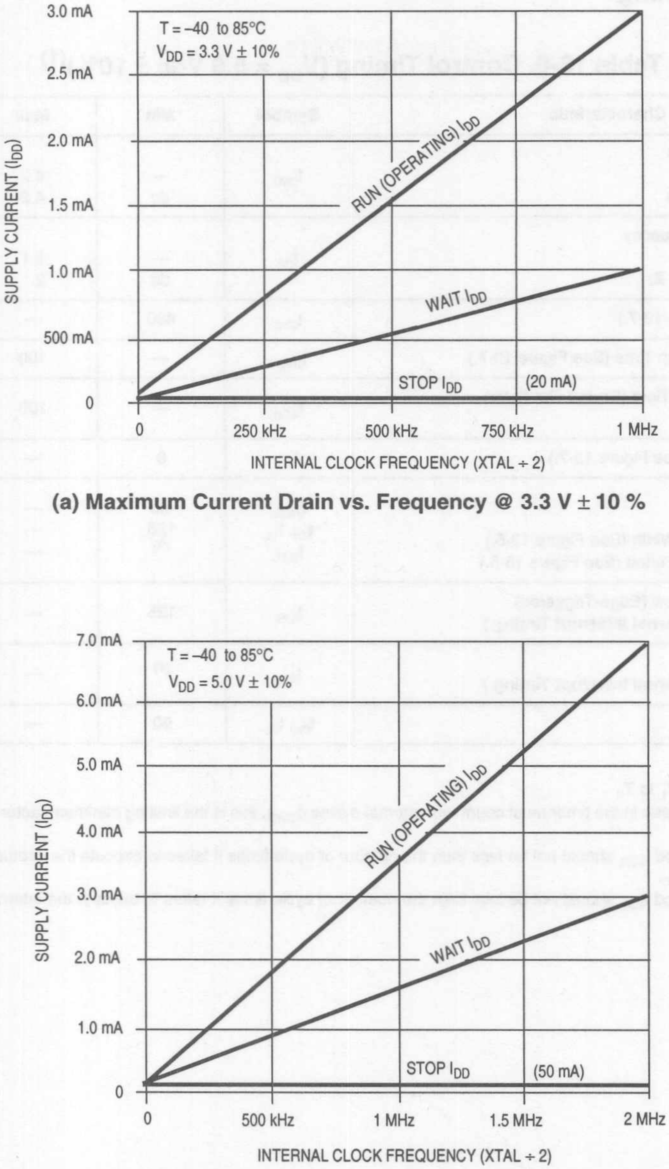
Figure 13-2. Typical Voltage Compared to Current (Continued)



(a) Wait Mode

(b) Run Mode

Figure 13-3. Typical Current vs. Internal Frequency for Run and Wait Modes



(a) Maximum Current Drain vs. Frequency @ 3.3 V ± 10 %

(b) Maximum Current Drain vs. Frequency @ 5 V ± 10%

Figure 13-4. Total Current Drain vs. Frequency

13.7 Control Timing

Table 13-6. Control Timing (V_{DD} = 5.0 Vdc ± 10%)[1]

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	f_{OSC}	— dc	4.2 4.2	MHz
Internal Operating Frequency Crystal (f_{OSC} ÷ 2) External Clock (f_{OSC} ÷ 2)	f_{OP}	— dc	2.1 2.1	MHz
Cycle Time (See Figure 13-7.)	t_{CYC}	480	—	ns
Crystal Oscillator Startup Time (See Figure 13-7.)	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (See Figure 13-6.)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width (See Figure 13-7.)	t_{RL}	8	—	t_{CYC}
Timer Resolution[2] Input Capture Pulse Width (See Figure 13-5.) Input Capture Pulse Period (See Figure 13-5.)	t_{RESL} t_{TH}, t_{TL} t_{TLTL}	4.0 125 [3]	— — —	t_{CYC} ns t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered) (See **Figure 4-2. External Interrupt Timing.**)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (See **Figure 4-2. External Interrupt Timing.**)	t_{ILIL}	[4]	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

NOTES:

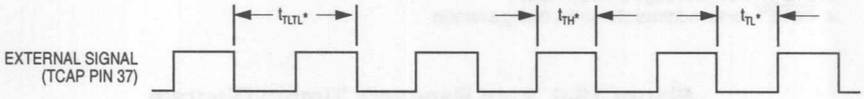
1. V_{SS} = 0 Vdc; $T_A = T_L$ to T_H
2. Since a 2-bit prescaler in the timer must count four internal cycles (t_{CYC}), this is the limiting minimum factor in determining the timer resolution.
3. The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.
4. The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.

Table 13-7. Control Timing (V_{DD} = 3.3 Vdc ± 0.3 Vdc)[1]

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	f_{OSC}	— dc	2.0 2.0	MHz
Internal Operating Frequency Crystal (f_{OSC} ÷ 2) External Clock (f_{OSC} ÷ 2)	f_{OP}	— dc	1.0 1.0	MHz
Cycle Time (See Figure 13-7.)	t_{CYC}	1000	—	ns
Crystal Oscillator Startup Time (See Figure 13-7.)	t_{OXOV}	—	100	ms
•Stop Recovery Startup Time (Crystal Oscillator) (See Figure 13-6.)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width, Excluding Power-Up (See Figure 13-7.)	t_{RL}	8	—	t_{CYC}
Timer Resolution[2] Input Capture Pulse Width (See Figure 13-5.) Input Capture Pulse Period (See Figure 13-5.)	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 250 [3]	 — — —	 t_{CYC} ns t_{CYC}
Interrupt Pulse Width Low (Edge-Triggered) (See **Figure 4-2. External Interrupt Timing.**)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (See **Figure 4-2. External Interrupt Timing.**)	t_{ILIL}	[4]	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

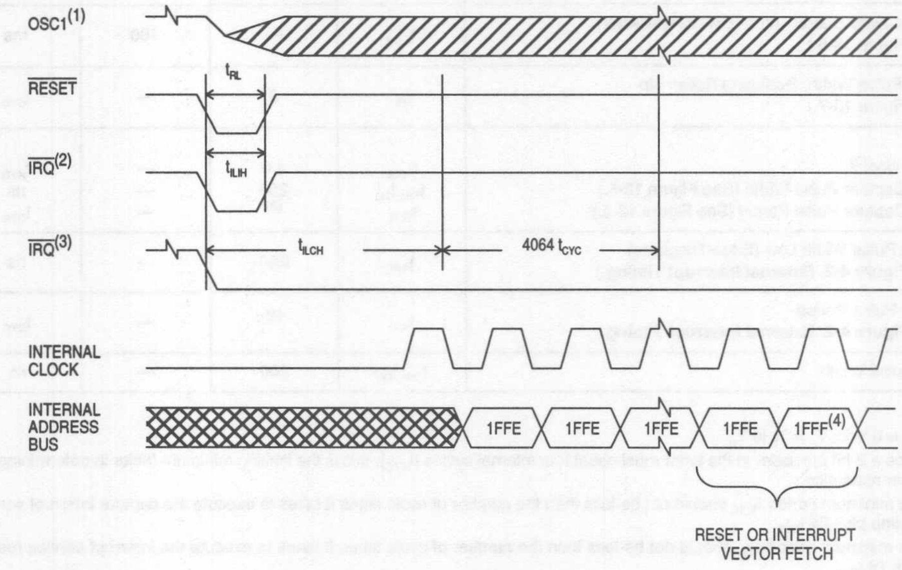
NOTES:

1. V_{SS} = 0 Vdc; $T_A = T_L$ to T_H
2. Since a 2-bit prescaler in the timer must count four internal cycles (t_{CYC}), this is the limiting minimum factor in determining the timer resolution.
3. The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.
4. The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.



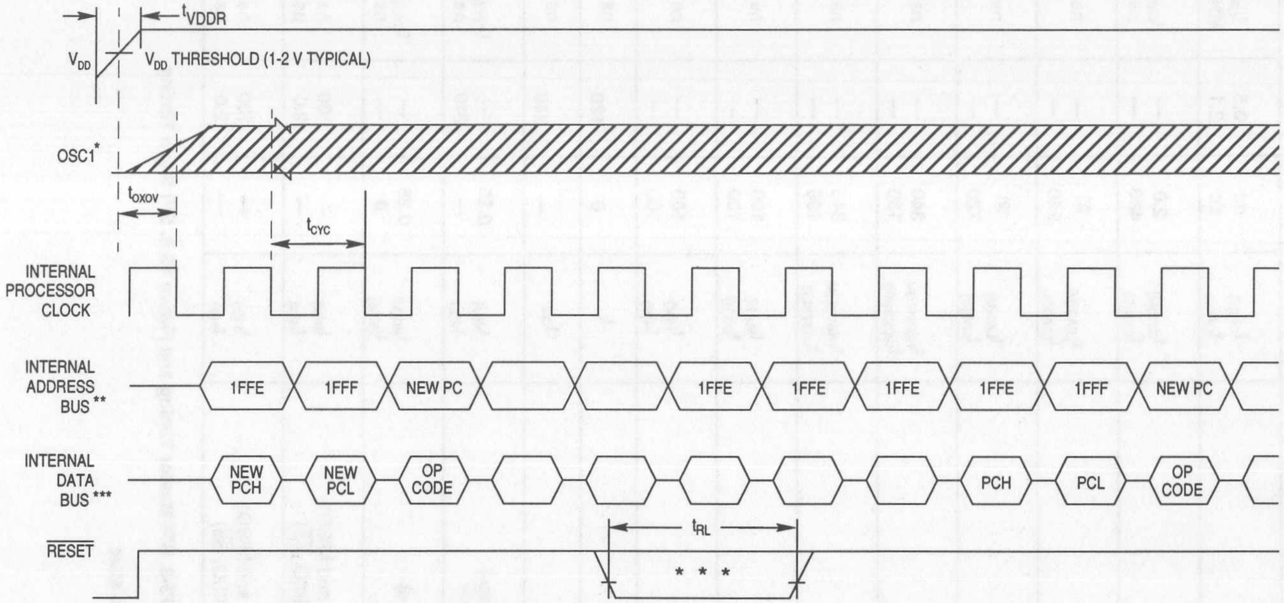
*Refer to timer resolution data in Table 13-6 and Table 13-7.

Figure 13-5. Timer Relationships



NOTES:
1. Represents the internal gating of the OSC1 pin
2. IRQ pin edge-sensitive option
3. IRQ pin level and edge-sensitive option
4. RESET vector address shown for timing example

Figure 13-6. Stop Recovery Timing Diagram



* OSC1 line is not meant to represent frequency. It is only used to represent time.

** Internal timing signal and bus information are not available externally.

***The next rising edge of the internal processor clock following the rising edge of RESET initiates the reset sequence.

Figure 13-7. Power-On Reset and External Reset Timing Diagram

Table 13-8. Serial Peripheral Interface (SPI) Timing (V_{DD} = 5.0 Vdc ± 10%)

Number[1]	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{OP(M)}$ $f_{OP(S)}$	 dc dc	 0.5 2.1	 f_{OP} MHz
1	Cycle Time Master Slave	 $t_{CYC(M)}$ $t_{CYC(S)}$	 2.0 480	 — —	 t_{CYC} ns
2	Enable Lead Time Master Slave	 $t_{LEAD(M)}$ $t_{LEAD(S)}$	 [2] 240	 — —	ns
3	Enable Lag Time Master Slave	 $t_{LAG(M)}$ $t_{LAG(S)}$	 [2] 720	 — —	ns
4	Clock (SCK) High Time Master Slave	 $t_{W(SCKH)M}$ $t_{W(SCKH)S}$	 340 190	 — —	ns
5	Clock (SCK) Low Time Master Slave	 $t_{W(SCKL)M}$ $t_{W(SCKL)S}$	 340 190	 — —	ns
6	Data Setup Time (Inputs) Master Slave	 $t_{SU(M)}$ $t_{SU(S)}$	 100 100	 — —	ns
7	Data Hold Time (Inputs) Master Slave	 $t_{H(M)}$ $t_{H(S)}$	 100 100	 — —	ns
8	Access Time[3] Slave	t_A	 0	 120	ns
9	Disable Time[4] Slave	t_{DIS}	 —	 240	ns
10	Data Valid Time Master (Before Capture Edge) Slave (After Enable Edge)[5]	 $t_{V(M)}$ $t_{V(S)}$	 0.25 —	 — 240	 $t_{CYC(M)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	 $t_{HO(M)}$ $t_{HO(S)}$	 0.25 0	 — —	 $t_{CYC(M)}$ ns
12	Rise Time[6] SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	 $t_{R(M)}$ $t_{R(S)}$	 — —	 100 2.0	 ns µs
13	Fall Time[7] SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	 $t_{F(M)}$ $t_{F(S)}$	 — —	 100 2.0	 ns µs

NOTES:

[1] Numbers refer to dimensions in **Figure 13-8. SPI Master Timing** and **Figure 13-9. SPI Slave Timing.**

[2] Signal production depends on software.

[3] Time to data active from high-impedance state.

[4] Hold time to high-impedance state.

[5] With 200 pF on all SPI pins.

[6] 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF.

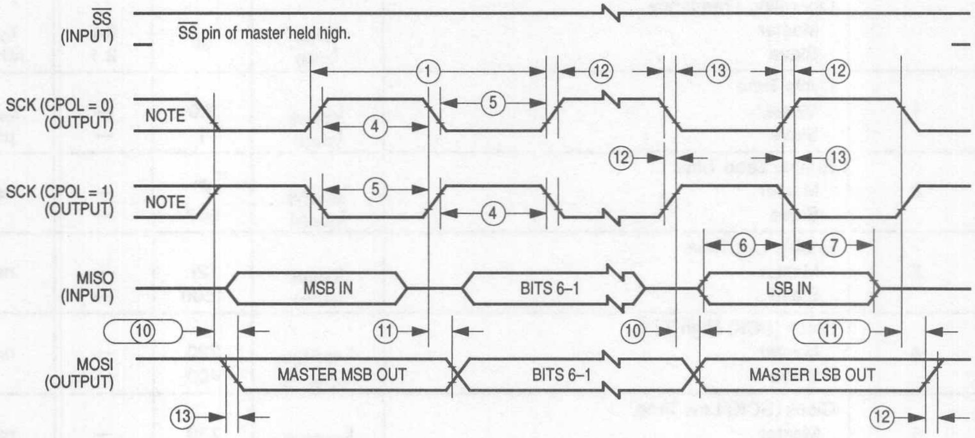
[7] 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF.

Table 13-9. Serial Peripheral Interface (SPI) Timing (V_{DD} = 3.3 Vdc ± 10%)

Number[1]	Characteristic	Symbol	Min	Max	Unit
—	Operating Frequency Master Slave	$f_{OP(M)}$ $f_{OP(S)}$	dc	0.5 2.1	f_{OP} MHz
1	Cycle Time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 1	— —	t_{CYC} μs
2	Enable Lead Time Master Slave	$t_{LEAD(M)}$ $t_{LEAD(S)}$	(2) 500	— —	ns
3	Enable Lag Time Master Slave	$t_{LAG(M)}$ $t_{LAG(S)}$	(2) 1500	— —	ns
4	Clock (SCK) High Time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	720 400	— —	ns
5	Clock (SCK) Low Time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	720 400	— —	ns
6	Data Setup Time (Inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	200 200	— —	ns
7	Data Hold Time (Inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	200 200	— —	ns
8	Access Time[3] Slave	t_A	0	250	ns
9	Disable Time[4] Slave	t_{DIS}	—	500	ns
10	Data Valid Time Master (Before Capture Edge) Slave (After Enable Edge)[5]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 500	$t_{CYC(M)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	$t_{HO(M)}$ $t_{HO(S)}$	0.25 0	— —	$t_{CYC(M)}$ ns
12	Rise Time[6] SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{R(M)}$ $t_{R(S)}$	— —	200 2.0	ns μs
13	Fall Time[7] SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	$t_{F(M)}$ $t_{F(S)}$	— —	200 2.0	ns μs

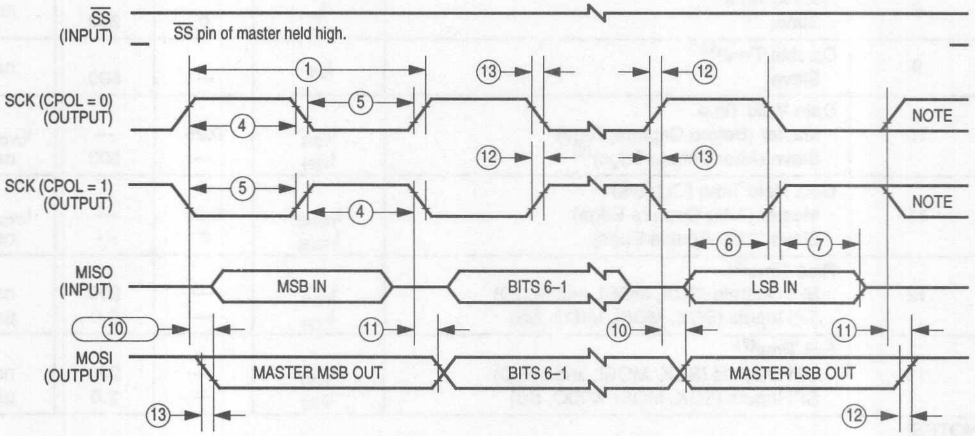
NOTES:

1. Numbers refer to dimensions in **Figure 13-8. SPI Master Timing** and **Figure 13-9. SPI Slave Timing.**
2. Signal production depends on software.
3. Time to data active from high-impedance state.
4. Hold time to high-impedance state.
5. With 200 pF on all SPI pins.
6. 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF.
7. 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF.



NOTE: This first clock edge is generated internally, but is not seen at the SCK pin.

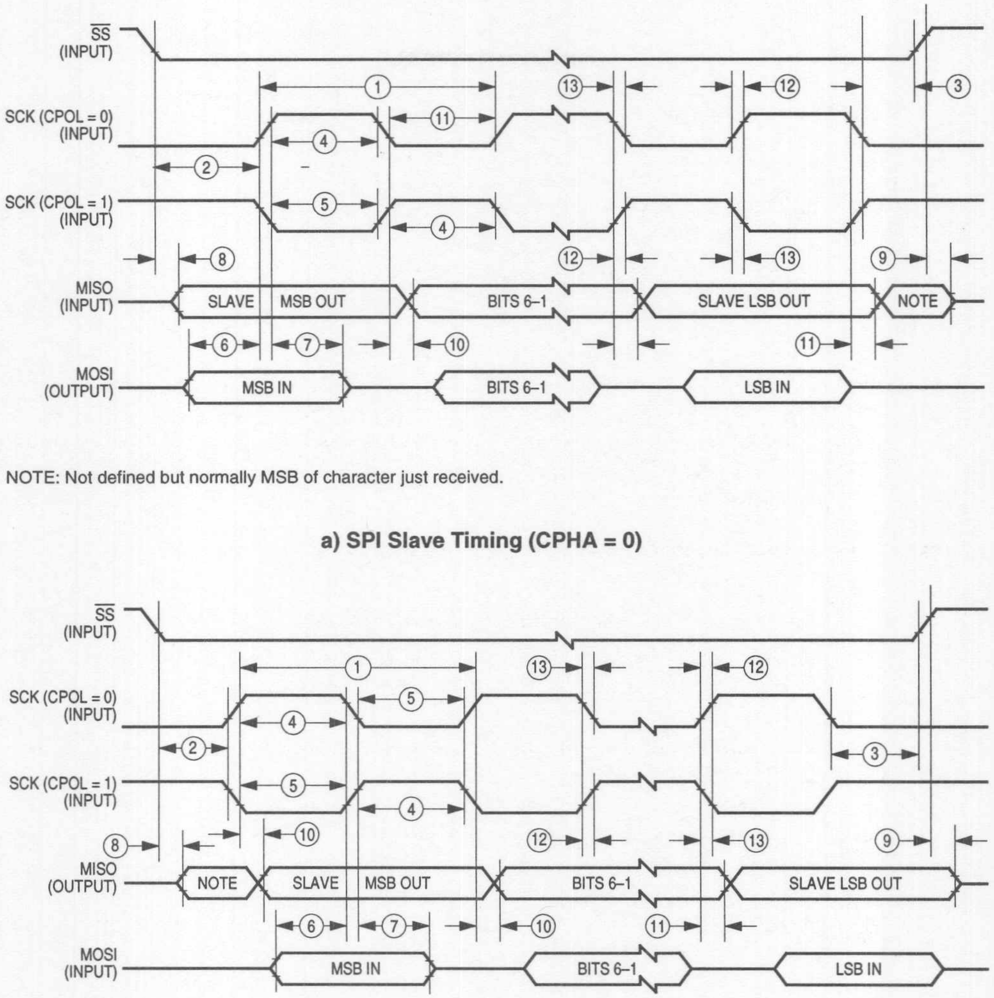
a) SPI Master Timing (CPHA = 0)



NOTE: This last clock edge is generated internally, but is not seen at the SCK pin.

b) SPI Master Timing (CPHA = 1)

Figure 13-8. SPI Master Timing



NOTE: Not defined but normally MSB of character just received.

a) SPI Slave Timing (CPHA = 0)

NOTE: Not defined but normally LSB of character previously transmitted.

b) SPI Slave Timing (CPHA = 1)

Figure 13-9. SPI Slave Timing

SECTION 14
MECHANICAL SPECIFICATIONS

14.1 Introduction

Package dimensions available at the time of this publication for the MC68HC705C4A are provided in this section. The packages are:

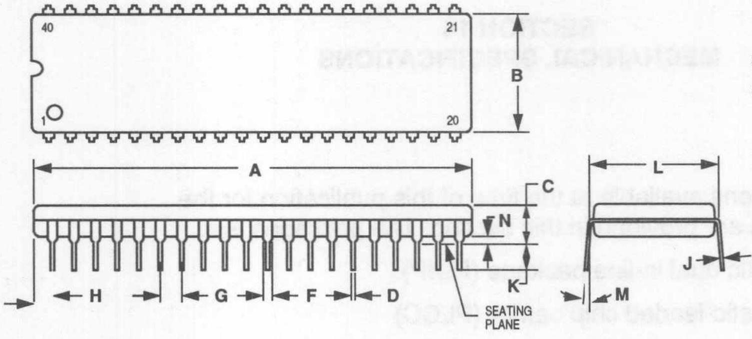
- 40-pin plastic dual in-line package (PDIP)
- 44-lead plastic-leaded chip carrier (PLCC)
- 44-pin quad flat pack (QFP)
- 42-pin shrink dual in-line package (SDIP)

To make sure that you have the latest case outline specifications, contact one of the following:

- Local Motorola Sales Office
- Motorola Mfax
  - Phone 602-244-6609
  - EMAIL rmfax0@email.sps.mot.com
- Worldwide Web (wwweb) at http://design-net.com

Follow Mfax or wwweb on-line instructions to retrieve the current mechanical specifications.

14.2 40-Pin Plastic Dual In-Line Package (PDIP)

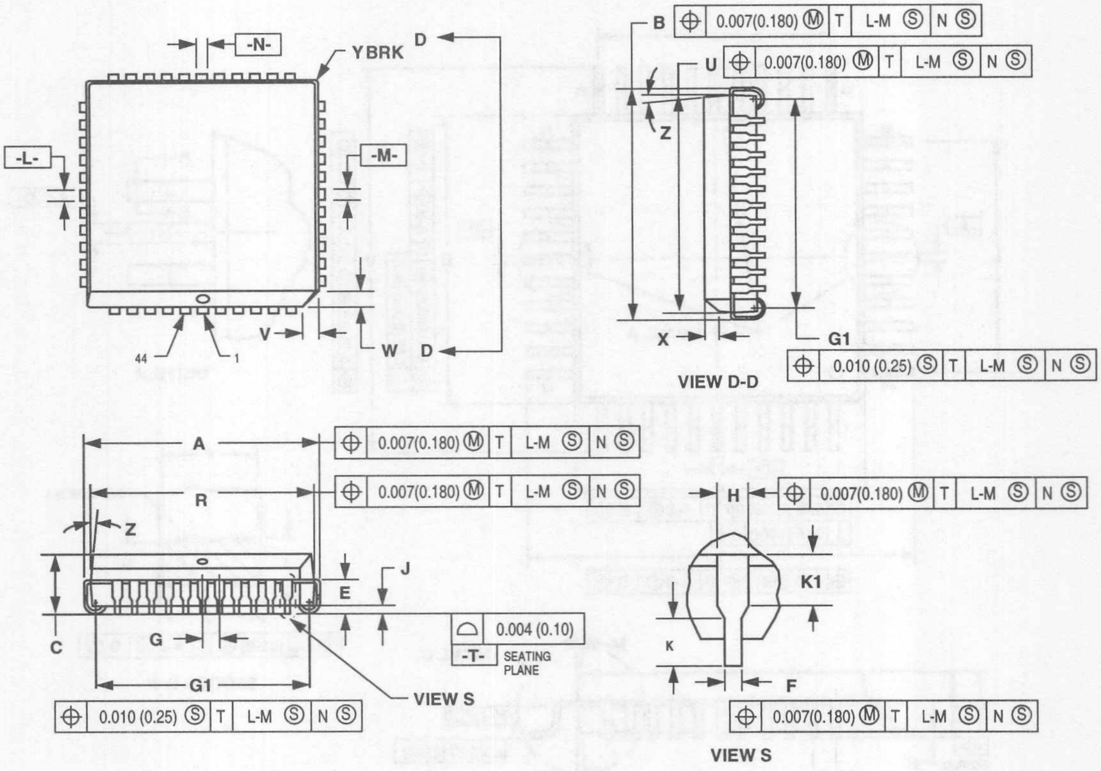


NOTES:
1. POSITION TOLERANCE OF LEADS (D), SHALL BEWITHIN 0.25 (0.010) AT MAXIMUM MATERIAL CONDITIONS, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	1°	0°	1°
N	0.51	1.02	0.020	0.040

Figure 14-1. MC68HC705C4AP Package Dimensions (Case #711)

14.3 44-Lead Plastic-Leaded Chip Carrier (PLCC)



-N-
YBRK
D
-L-
-M-
44
1
V
W
D
B
⊕ 0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
U
⊕ 0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
Z
X
G1
⊕ 0.010 (0.25) Ⓢ T L-M Ⓢ N Ⓢ
VIEW D-D
A
⊕ 0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
R
⊕ 0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
Z
J
E
C
G
0.004 (0.10)
-T-
SEATING PLANE
G1
⊕ 0.010 (0.25) Ⓢ T L-M Ⓢ N Ⓢ
VIEW S
H
⊕ 0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
K1
K
F
⊕ 0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
VIEW S

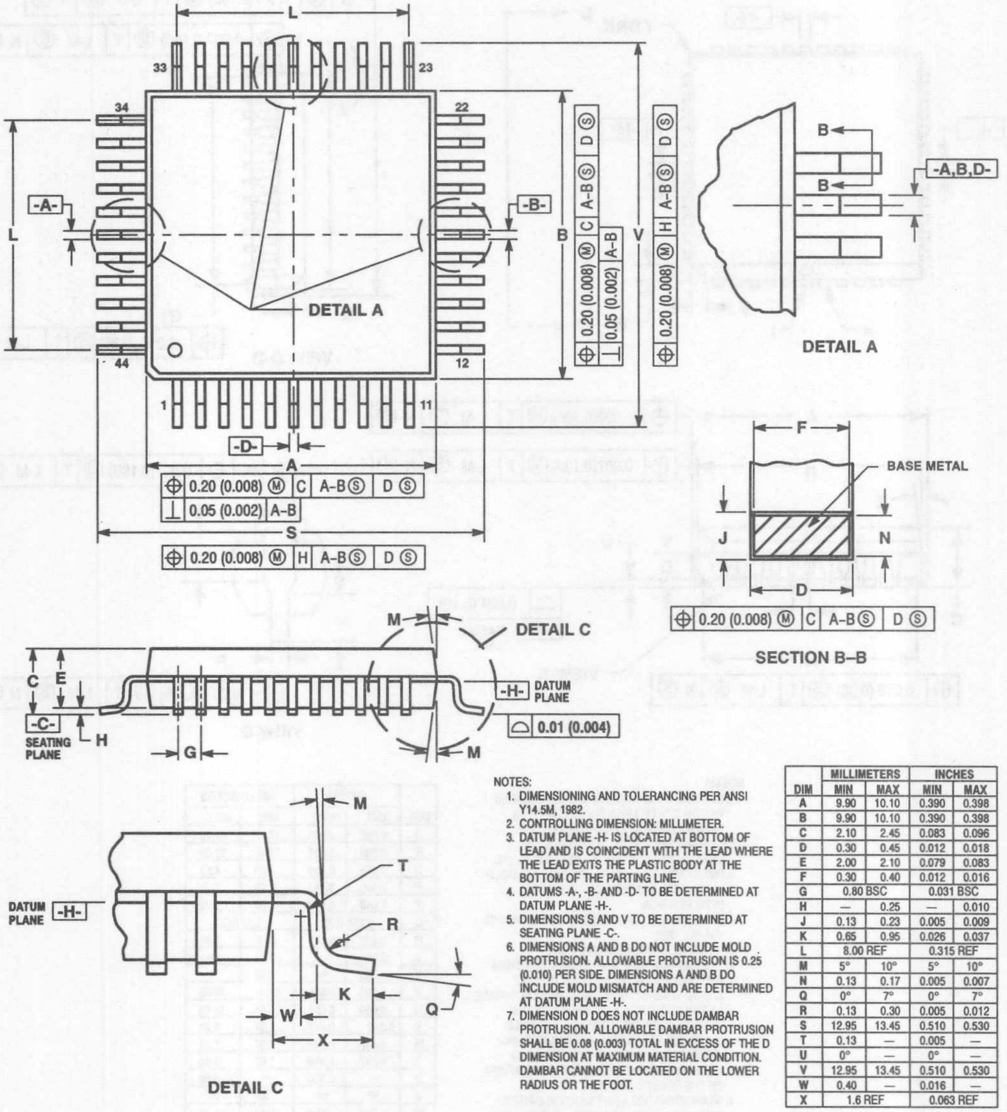
NOTES:

1. DATUMS -L-, -M-, AND -N- ARE DETERMINED WHERE TOP OF LEAD SHOLDERS EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIMENSION G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIMENSION R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.25) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF THE MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTTOM OF THE PLASTIC BODY.
7. DIMINSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTUSION(S) SHALL NOT CAUSE THE H DIMINSION TO BE GREATER THAN 0.037 (0.9406). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMINISION TO SMALLER THAN 0.025 (0.635).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.685	0.695	17.40	17.65
B	0.685	0.695	17.40	17.65
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	0.650	0.656	16.51	16.66
U	0.650	0.656	16.51	16.66
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	0.610	0.630	15.50	16.00
K1	0.040	—	1.02	—

Figure 14-2. MC68HC705C4AFN Package Dimensions (Case #777)

14.4 44-Pin Quad Flat Pack (QFP)



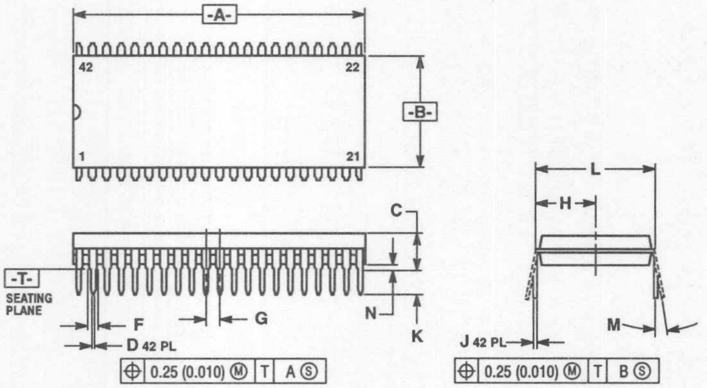
NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.90	10.10	0.390	0.398
B	9.90	10.10	0.390	0.398
C	2.10	2.45	0.083	0.096
D	0.30	0.45	0.012	0.018
E	2.00	2.10	0.079	0.083
F	0.30	0.40	0.012	0.016
G	0.80 BSC		0.031 BSC	
H	—	0.25	—	0.010
J	0.13	0.23	0.005	0.009
K	0.65	0.95	0.026	0.037
L	8.00 REF		0.315 REF	
M	5°	10°	5°	10°
N	0.13	0.17	0.005	0.007
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	12.95	13.45	0.510	0.530
T	0.13	—	0.005	—
U	0°	—	0°	—
V	12.95	13.45	0.510	0.530
W	0.40	—	0.016	—
X	1.6 REF		0.063 REF	

Figure 14-3. MC68HC705C4AFB Package Dimensions (Case #824E)

14.5 42-Pin Shrink Dual In-Line Package (SDIP)



NOTES:

1. DIMENSIONS AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSIONS A AND B DO NOT INCLUDE MOLD FLASH. MAXIMUM MOLD FLASH 0.25 (0.010).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.435	1.465	36.45	37.21
B	0.540	0.560	13.72	14.22
C	0.155	0.200	3.94	5.08
D	0.014	0.022	0.36	0.56
F	0.032	0.046	0.81	1.17
G	0.070 BSC		1.778 BSC	
H	0.300 BSC		7.62 BSC	
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.600 BSC		15.24 BSC	
M	0°	15°	0°	15°
N	0.020	0.040	0.51	1.02

Figure 14-4. MC68HC705C4AB Package Dimensions (Case #858)

SECTION 15
ORDERING INFORMATION

15.1 Introduction

This section contains ordering information for the available package types.

15.2 MCU Order Numbers

Table 15-1 lists the MC order numbers.

Table 15-1. MC68HC705C4A Order Numbers

Package Type	Temperature Range	Order Number
40-Pin Plastic Dual In-Line Package (PDIP)	–40 °C to +85 °C	MC68HC705C4AC(1)P(2)
44-Lead Plastic-Leaded Chip Carrier (PLCC)	–40 °C to +85 °C	MC68HC705C4ACFN(3)
44-Lead Quad Flat Pack (QFP)	–40 °C to +85 °C	MC68HC705C4ACFB(4)
42-Pin Shrink Dual In-Line Package (SDIP)	–40 °C to +85 °C	MC68HC705C4ACB(5)

NOTES:
1. C = Extended temperature range (–40 °C to +85 °C).
2. P = Plastic dual in-line package (PDIP)
3. FN = Plastic-leaded chip carrier (PLCC)
4. FB = Quad flat pack (QFP)
5. B = Shrink dual in-line package (SDIP)

15

APPENDIX A
MC68HSC705C4A

A.1 Introduction

The MC68HSC705C4A is an enhanced, high-speed version of the MC68HC705C4A, featuring a 4-MHz bus speed.

The data in *MC68HC705C4A Technical Data* applies to the MC68HSC705C4A with the exceptions given in this appendix.

A.2 DC Electrical Characteristics

Table A-1. High-Speed DC Electrical Characteristics (V_{DD} = 5.0 V ± 10%)

Characteristic	Symbol	Min	Typ[1]	Max	Unit
Output High Voltage					
I_{LOAD} = –0.8 mA					
PA7–PA0, PB7–PB0, PC6–PC0, TCMP			—	—	
I_{LOAD} = –1.6 mA	V_{OH}	V_{DD} – 0.8			V
PD4–PD1			—	—	
I_{LOAD} –5.0 mA					
PC7			—	—	
Output Low Voltage					
I_{LOAD} = 1.6 mA					
PA7–PA0, PB7–PB0, PC6–PC0, PD4–PD1	V_{OL}	—	—	0.4	V
I_{LOAD} = 20 mA					
PC7		—	—	0.4	
Supply Current[2]					
Run[3]		—	5.92	14	mA
WAIT[4]		—	2.27	7.0	mA
STOP[5]	I_{DD}				
25 °C		—	5	50	µA
–40 °C to +85 °C		—	2.0	50	µA

NOTES:

1. Typical values reflect average measurements at midpoint of voltage range at 25 °C.
2. I_{DD} measured with port B pullup devices disabled.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 8.0 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. OSC2 capacitance linearly affects Run I_{DD}.
4. WAIT I_{DD} measured using external square wave clock source (f_{OSC} = 8.0 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V. All ports configured as inputs. SPI and SCI disabled. If SPI and SCI enabled, add 10% current draw. OSC2 capacitance linearly affects WAIT I_{DD}.
5. STOP I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs. V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V.

Table A-2. High-Speed DC Electrical Characteristics (V_{DD} = 3.3 V ± 10%)

Characteristic	Symbol	Min	Typ[1]	Max	Unit
Output High Voltage I_{LOAD} = –0.2 mA PA7–PA0, PB7–PB0, PC6–PC0, TCMP I_{LOAD} = –0.4 mA PD4–PD1 I_{LOAD} = –1.5 mA PC7	V_{OH}	V_{DD} – 0.3	— — —	— — —	V
Output Low Voltage I_{LOAD} = 0.4 mA PA7–PA0, PB7–PB0, PC6–PC0, PD4–PD1 I_{LOAD} = 6.0 mA PC7	V_{OL}	— —	— —	0.3 0.3	V
Supply Current[2] Run[3] WAIT[4] STOP[5]	I_{DD}	— — —	1.91 0.915 2.0	6.0 2.0 20	mA mA µA

NOTES:

1. Typical values reflect average measurements at midpoint of voltage range at 25 °C.
2. I_{DD} measured with port B pullup devices disabled.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. OSC2 capacitance linearly affects Run I_{DD}.
4. WAIT I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20pF on OSC2. V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V. All ports configured as inputs. SPI and SCI disabled. If SPI and SCI enabled, add 10% current draw. OSC2 capacitance linearly affects WAIT I_{DD}.
5. STOP I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs. V_{IL} = 0.2 V; V_{IH} = V_{DD} – 0.2 V.

A.3 Control Timing

Table A-3. High-Speed Control Timing (V_{DD} = 5.0 V ± 10%)

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency Crystal Oscillator External Clock	f_{OSC}	— dc	8.0 8.0	MHz
Internal Operating Frequency ($f_{OSC} \div 2$) Crystal Oscillator External Clock	f_{OP}	— dc	4.0 4.0	MHz
Cycle Time	t_{CYC}	250	—	ns
Input Capture Pulse Width	t_{TH}, t_{TL}	65	—	ns
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	65	—	ns
OSC1 Pulse Width	t_{OH}, t_{OL}	45	—	ns

Table A-4. High-Speed Control Timing (V_{DD} = 3.3 V ± 10%)

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency Crystal Oscillator External Clock	f_{OSC}	— dc	4.0 4.0	MHz
Internal Operating Frequency ($f_{OSC} \div 2$) Crystal Oscillator External Clock	f_{OP}	— dc	2.0 2.0	MHz
Cycle Time	t_{CYC}	476	—	ns
Input Capture Pulse Width	t_{TH}, t_{TL}	125	—	ns
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

Table A-5. High-Speed SPI Timing (V_{DD} = 5.0 V ± 10%)

Diagram Number[1]	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	$f_{OP(S)}$ $f_{OP(S)}$	dc dc	0.5 4.0	f_{OP} MHz
1	Cycle Time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 250	— —	t_{CYC} ns
2	Enable Lead Time Master Slave	$t_{LEAD(M)}$ $t_{LEAD(S)}$	NOTE [2] 125	— —	ns
3	Enable Lag Time Master Slave	$t_{LAG(M)}$ $t_{LAG(S)}$	NOTE[2] 375	— —	ns
4	Clock (SCK) High Time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	170 95	— —	ns
5	Clock (SCK) Low Time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	170 95	— —	ns
6	Data Setup Time (Inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	50 50	— —	ns
7	Data Hold Time (Inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	50 50	— —	ns
8	Access Time[3] Slave	t_A	0	60	ns
9	Disable Time[4] Slave	t_{DIS}	—	120	ns
10	Data Valid Time Master (Before Capture Edge) Slave (After Enable Edge)[5]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 120	$t_{CYC(M)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	$t_{HO(M)}$ $t_{HO(S)}$	0.25 0	— —	$t_{CYC(M)}$ ns
12	Rise Time[6] SPI Outputs (SCK, MOSI, and $\overline{\text{MISO}}$) SPI Inputs (SCK, MOSI, MISO, $\overline{\text{SS}}$)	t_{RM} t_{RS}	— —	50 2.0	ns µs
13	Fall Time[7] SPI Outputs (SCK, MOSI, and $\overline{\text{MISO}}$) SPI Inputs (SCK, MOSI, MISO, $\overline{\text{SS}}$)	t_{FM} t_{FS}	— —	50 2.0	ns µs

NOTES:
1. Diagram numbers refer to dimensions in **Figure 13-8. SPI Master Timing** and **Figure 13-9. SPI Slave Timing**.
2. Signal production depends on software.
3. Time to data active from high-impedance state.
4. Hold time to high-impedance state.
5. With 200 pF on all SPI pins.
6. 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF.
7. 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF.

Table A-6. High-Speed SPI Timing (V_{DD} = 3.3 V ± 10%)

Diagram Number[1]	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	$f_{OP(S)}$ $f_{OP(S)}$	dc dc	0.5 2.1	f_{OP} MHz
1	Cycle Time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 480	— —	t_{CYC} ns
2	Enable Lead Time Master Slave	$t_{LEAD(M)}$ $t_{LEAD(S)}$	NOTE [2] 240	— —	ns
3	Enable Lag Time Master Slave	$t_{LAG(M)}$ $t_{LAG(S)}$	NOTE[2] 720	— —	ns
4	Clock (SCK) High Time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	340 190	— —	ns
5	Clock (SCK) Low Time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	340 190	— —	ns
6	Data Setup Time (Inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	100 100	— —	ns
7	Data Hold Time (Inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	100 100	— —	ns
8	Access Time[3] Slave	t_A	0	120	ns
9	Disable Time[4] Slave	t_{DIS}	—	240	ns
10	Data Valid Time Master (Before Capture Edge) Slave (After Enable Edge)[5]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 240	$t_{CYC(M)}$ ns
11	Data Hold Time (Outputs) Master (After Capture Edge) Slave (After Enable Edge)	$t_{HO(M)}$ $t_{HO(S)}$	0.25 0	— —	$t_{CYC(M)}$ ns
12	Rise Time[6] SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	t_{RM} t_{RS}	— —	100 2.0	ns µs
13	Fall Time[7] SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	t_{FM} t_{FS}	— —	100 2.0	ns µs

NOTES:

1. Diagram numbers refer to dimensions in **Figure 13-8. SPI Master Timing** and **Figure 13-9. SPI Slave Timing.**
2. Signal production depends on software.
3. Time to data active from high-impedance state.
4. Hold time to high-impedance state.
5. With 200 pF on all SPI pins.
6. 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF.
7. 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF.

A.4 Ordering Information

Table A-7 provides ordering information for the MC68HSC705C4A.

Table A-7. MC68HSC705C4A Order Numbers

Package Type	Temperature Range	Order Number
40-Pin Plastic Dual In-Line Package (PDIP)	–40 °C to +85 °C	MC68HSC705C4AC(1)P(2)
44-Lead Plastic-Leaded Chip Carrier (PLCC)	–40 °C to +85 °C	MC68HSC705C4ACFN(3)
44-Lead Quad Flat Pack (QFP)	–40 °C to +85 °C	MC68HSC705C4ACFB(4)
42-Pin Shrink Dual In-Line Package (SDIP)	–40 °C to +85 °C	MC68HSC705C4ACB(5)

NOTES:
1. C = Extended temperature range (–40 °C to +85 °C)
2. P = Plastic dual in-line package (PDIP)
3. FN = Plastic-leaded chip carrier (PLCC)
4. FB = Quad flat pack (QFP)
5. B = Shrink dual in-line package (SDIP)

INDEX